Prior Knowledge and Statistical Models of Learning

Lewis Andrew Bott

Thesis submitted in part fulfillment for the degree of Doctor of Philosophy

Department of Psychology
University of Warwick

March 2001

Table of Contents

List of Figures

List of Tables

Acknowledgements

First, I would like to express my gratitude to my supervisors, Evan Heit and Gordon Brown, for their invaluable guidance and advice throughout the course of this research. I am also indebted to Neil and Eoghan for writing some of the C programs for me. Thanks also to everybody at Warwick for their friendship and help whenever I've needed it. Finally, I thank Kat and my family for their love and encouragement.

Declaration

The modelling work presented in Chapter 3, Section 3.1.2, has been published in Heit and Bott (2000). This, and all the work reported in the thesis is my own work. The thesis has not been submitted for a degree at another university.

Abstract

The research reported here describes the effects of prior knowledge on how people form categories and learn continuous mappings. Chapter 2 is a review of the past research on knowledge effects in the statistical and psychological literature. Chapter 3 presents simulations of a set of experiments carried out by Heit and Bott (2000) into how knowledge is selected in a category learning task. The model was shown to account for the results of Heit and Bott and generate several new predictions concerning blocking effects with the use of prior knowledge. However, empirical testing of these predictions failed to demonstrate these effects. Chapter 4 describes work testing Delosh, McDaniel and Busemeyer's (1997) model of function learning, the Extrapolation Associative Learning Model (EXAM). Experiments were carried out demonstrating that a model that assumes only linear extrapolation, such as EXAM, is inadequate as a generic model of function learning. An alternative model to EXAM is presented which is constructed of several components, each module applying different quantities of prior knowledge to the task. Chapter 5 presents experiments investigating the extent to which participants abstract and apply functions in transfer-tasks. The results demonstrate that models of function learning must be able to restrict their range of allowable solutions in psychologically plausible ways.

Chapter 1

The notion of 'learning' is generally thought of as the process of acquiring *new* information to satisfy some goal. Less often considered is the role played by the knowledge the organism already possesses, or its *prior knowledge* of the task at hand. Furthermore, knowledge does not appear to take a passive role in the learning of new concepts; rather, there seems to be a constant drive to relate new ideas to old and for this knowledge to shape the learning process in general.

For example, imagine you were visiting a foreign city for the first time and you were interested in identifying the architectural styles. Your knowledge would guide you on which buildings to examine, which attributes of the buildings to pay attention to, or how to group the buildings together to relate them to known architectural styles. All of these provide benefits to the learning process in terms of the reduced time needed to acquire a concept and improved generalisation performance. To emphasise this point, consider what the task would be like with minimal prior knowledge applied: you might try to group buildings based on the colour of the doors, or whether they're near a bus stop or not, or be struck by the correlation between the number of windows and the height of the buildings. These attributes of the environment are all potentially relevant but, from the point of view of someone with knowledge of architecture, exceedingly unlikely indicators of architectural style.

To take another example, consider learning a second language, say Spanish. If you already speak French, then the task appears far easier than if you speak only

German. This is because more of the vocabulary and grammatical structure of French can be mapped one-to-one than German. On the other hand, you may suffer some negative effects of your knowledge if you are reluctant to give up your French, through 'false friends', or incorrect pronunciations etc. There is always a balance to be struck when applying prior knowledge: benefits can certainly occur, but sticking too rigidly to what one knows risks missing the target concept altogether.

In short, our prior knowledge influences category formation in a wide variety of ways. This in turn means that if we are to achieve some understanding of categorisation, some investigation of the effects of prior knowledge is needed. On the other hand, one could argue that it is sensible to start off investigating the empirical component of learning, and only when a thorough understanding of these processes has been achieved, should we move on to investigating the effects of knowledge. After all, if we can eliminate the effects of participants' knowledge from our experiments, far less variation will occur in the responses. This has been the research strategy of the vast majority of cognitive scientists: witness the wealth of relatively successful models of association or perceptual categorisation (Gluck & Bower, 1988; Kruschke, 1992; Nosofsky, 1986; Pearce, 1987; Rescorla & Wagner, 1972; among others). So, why study the relatively high-level effects of prior knowledge? First, the empirical models seem to have reached a plateau in what they can explain. They have been sufficiently successful in accounting for more or less all the possibilities within the domain of abstract stimuli but their scope needs expanding and an obvious direction is towards research into the effects of prior knowledge. Secondly, it may not be

possible to consider the two aspects separately: prior knowledge may alter the *processes* by which we acquire new information. If this is the case, then we cannot simply tag on research into the effects of prior knowledge; the two must be developed concurrently.

This thesis is an attempt to redress some of the imbalance between research into data-driven learning and research into the effects of prior knowledge. The methodology adopted involves both quantitative modelling techniques and traditional experimental psychology. While the experiments allow us to investigate the truth of our hypotheses directly, the modelling provides much needed theoretical support for generating those hypotheses, as well as a mechanism for incorporating insights from other disciplines. Indeed, one of the aims of the thesis is to examine how the concept of prior knowledge in probability theory and statistics can be incorporated into current models of category formation.

Although the effects of knowledge are clearly important in all areas of psychology, we selected the sub-disciplines of categorisation and regression (the latter generally known as function learning) for investigation. There were three reasons for this choice. First, a large proportion of psychology can be construed in terms of these processes and the findings should thus be widely applicable. For example, object recognition can be thought of as the process of allocating an image to its appropriate category, or throwing and catching as learning the functions which characterise the path of moving objects, or language acquisition as the assignment of category labels to situations in the environment. Secondly,

modern statistics has devoted a lot of effort to discovering good algorithms for classification and regression. It seems sensible to try to investigate whether its findings on prior knowledge are applicable to psychology. Finally, research into categorisation and function learning has begun to investigate how knowledge interacts with data, thus providing a platform on which the present research can build.

The next section discusses current models of categorisation and function learning and the work that has been done so far on prior knowledge, not with aim of providing a detailed exposition - this will form the basis of later chapters – but to illustrate how difficult it is in general for these models to provide adequate explanations of prior knowledge effects.

Prior knowledge and models of function learning and categorisation

Several different representational formats are used by models of categorisation and function learning, including similarity-based (for example, Delosh, Busemeyer, & McDaniel, 1997; Medin & Shaffer, 1978; Nosofsky, 1986, Rosch & Mervis, 1975), feature-based (Tversky, 1977) or rule-based (Allen & Brooks, 1991; Brehmer, 1974). Because the current fashion is for similarity models, this section will use these as an example. Note however, that many of the comments made here apply equally well to the other formats.

According to similarity-based approaches, objects are represented on a multi-dimensional space where each dimension represents an attribute of interest. In function learning, examples are typically coded on only one or two dimensions, while categorisation models assume a space with perhaps four or five dimensions. The similarity of one object to another is determined by the output of a suitably defined distance function on the space. In categorisation, the decision about how to classify a new object depends on the similarity of that object to the items in the category of interest. For example, take a standard exemplar-based view (Nosofsky, 1986) of how you might classify a previously unseen dog into the appropriate species. The representations of dogs from various different classes are stored in memory, together with their species label, be it alsation, poodle, dachshund or whatever. The relevant dimensions here might be colour of fur, overall size, and propensity for inflecting harm, with each example of past dogs having values on these dimensions. On encountering the to-be-classified dog, the similarity (or distance) between its representation and each of the past dogs is assessed, and assignment is made to the category with the largest summed similarity to the test item.

An analogous situation in function learning might be to predict the length of time a machine will operate on, after being given a certain quantity of petrol. Previous examples of the relationship are stored as the quantity of petrol given to the machine with an associated length of operation. To obtain the predicted length of time for a new volume of petrol, an exemplar-based strategy (such as the Associative Learning Model of Busemeyer, Byun, Delosh, & McDaniel, 1997) would assume that the previously encountered time scores would be

summed and weighted by the similarity between the new petrol amount and old petrol amounts.

Even from this brief description of similarity models, several problems immediately stand out (see Murphy & Medin, 1985, and Hahn & Chater, 1997, for reviews). First is the question of determining the 'relevant' dimensions to compute similarity on. Objects potentially have an infinite number of attributes and only very few of them provide the information useful for classifying the object. The simplest way of understanding this is to note that there are times when the *lack* of an attribute is important in the classification. For example, if a building *does not* have windows, then it is unlikely to be an office block. Taking all dimensions into account means that that any two objects are maximally similar, through their shared not-features: they will both be similar in the sense that they are not blue, don't have spires, were not born on the 3rd of July, etc. A further problem with high dimensional spaces are the prohibitively large time and space requirements and the number of examples required to successfully shatter them (Bellman, 1961), an issue which will be dealt with in detail in Chapter 2. In short, an *a priori* method of specifying the relevant attributes is needed. Perceptual constraints can do some of the work (e.g. Goldstone, Steyvers, Spencer-Smith, & Kersten, 1999), but it is clear that knowledge carried from one task to another also guides the choice of dimensions (e.g. Pazzini, 1991).

Similarity-based models also fail to provide answers to why some categories or functions are learnt more quickly than others, or equivalently, why generalisation

for some categories is better than for others after a given period of time. For instance, if examples in the to-be-learnt category are normally distributed, then participants find it easier to learn these than if they are binomially distributed (Flannagan, Fried, & Holyoak, 1986). Another example is provided by Murphy and Allopena (1994), who showed that participants find tasks much easier if the to-be-learnt category distinction maps onto some previously known category. Furthermore, Pazzini (1991) demonstrated that using knowledge-based categories can reverse the usual ordering of difficulty: a non-linearly separable task is learnt more easily than a linearly separable one, if appropriate knowledge is invoked.

These are just a few of the questions the extant models of category and function learning do not address. Others abound, such as how the notion of causation might be incorporated into the models (Murphy & Medin, 1985), or how different categories are combined (Hampton, 1997), or why some exemplars seem to carry more weight than others (Heit, 1998). The point is not that these models are *incompatible* with knowledge effects, but that they don't provide any *explanation*. For example, many similarity-based models allow weights to be placed on the different dimensions, thus allowing the implementation of *a priori* determined dimensional importance. But this doesn't tell us why those dimensions were chosen in the first place: is there a general hierarchy of dimension sampling? How much do other learnt categories influence this choice? At what point are new dimensions sampled? It is possible to argue that these are secondary phenomena, in the sense that a sufficiently powerful learning mechanism will 'figure-out' the relevant learning criteria, as was the hope with

non-parametric algorithms like neural networks, but, as Chapter 2 will argue, the evidence is against this. Indeed, several researchers have suggested that the data-driven side to learning is almost trivial – deciding which and how much knowledge to incorporate is the process that needs explaining (Geman, Bienenstock, & Doursat, 1995; Minsky & Papert, 1969).

Overview of the thesis

As discussed above, the aims of the thesis is to investigate how prior knowledge influences learning and how current models might be developed to account for these effects. Chapter 2 combines perspectives from statistics, engineering and psychology to produce an interdisciplinary review of these ideas. The chapter formally introduces classification and describes why prior knowledge is needed to solve 'interesting' problems. After this, techniques for incorporating knowledge are discussed, under a distinction between knowledge used for its *information* value and that used for its *complexity* value.

Chapter 3 presents a computational model of a set of experiments described in Heit and Bott (2000). These experiments demonstrated how knowledge can have an increasing effect on performance in a concept learning task. The model shows that, by incorporating the influence of multiple hypotheses (or known categories), instead of a simple data-driven algorithm, simulation results capture the principal effects observed in Heit and Bott. Several novel predictions are generated from the model, which are then empirically tested.

Chapters 4 and 5 involve a change from categorisation to regression, or function learning. Chapter 4 examines how participants choose to extrapolate beyond the training data. In particular, the question of whether they pickup patterns in the data is examined, and whether they choose to apply these patterns in their generalisation responses. The question is important because purely data-driven models are unable to predict any but the simplest patterns in generalisation. Chapter 5 continues with this line of research by investigating how participants might transfer knowledge from one stage to another in a learning task. This allows some of the ideas discussed in Chapter 2 to be put into practice, by seeing how extant models of function learning predict the transfer effects. Finally, Chapter 6 summarises the findings and concludes the thesis.

Chapter 2

Categorisation experiments have traditionally involved completing highly abstract, laboratory-based tasks, such as learning to classify a series of geometric shapes as belonging to one or another category. One of the aims of the methodology has been to reduce the effects of the knowledge a participant may bring, thereby tapping the underlying processes without invoking large amounts of individual variation. This strategy has proved remarkably successful in providing detailed, quantitative models which have reached the stage where they can be used as tools of analysis to investigate other psychological phenomena (see Lamberts, 1995; Lamberts & Shapiro, in press; Nosofsky & Zaki, 1998).

Not surprisingly, these models have been heavily empirically driven so that knowledge outside the current context is not considered. For example, according to a prototype model (e.g. Homa, 1984; Posner & Keele, 1968, 1970), a novel item would be classified as a belonging to Category A if it is sufficiently similar to the prototype of the experimentally presented items labelled as 'A'. To obtain good quantitative fits, there has been no need to take account of items beyond those presented in the laboratory, nor of any other biases the participants may have on entering the experiment. However, the very methodology which made the models so quantitatively successful has meant that they seem far removed from the categorisation problems that occur beyond the laboratory. This has encouraged criticisms of these highly data-driven approaches from both psychological perspectives (Murphy & Medin, 1985; Pinker & Prince, 1988; Schyns, Goldstone, & Thibaut, 1998), and statistical ones (Minsky & Papert,

1969; Geman, Bienenstock, & Doursat, 1992; Frasconi, Gori, & Soda, 1995). The upshot has been that static, similarity-based models (such as those by Ashby & Gott, 1988; Medin & Schaffer, 1978; Nosofsky 1986; Rosch & Mervis, 1975, and many others) are considered inadequate to explain interesting aspects of categorisation. For example, Murphy and Medin point out that these theories provide no explanations for which attributes are taken into account when making a categorisation decision, while Geman *et al.* demonstrate that there is an insufficient quantity of information in the environment for such non-parametric classifiers to generalise. Finally, there has been a wealth of experimental evidence showing that prior knowledge can have a dramatic effect on the categorisation process, such as Pazzini's (1991) demonstration that whether a linear or non-linear problem is learnt first depends on the cover story given to participants (and see Heit, 1997, for a review).

Despite these criticisms, there has been a reluctance to try to incorporate knowledge effects into formal models of psychological categorisation. Part of the problem is that it has been difficult to identify what 'prior knowledge' is and why it is needed – there seems to be no framework to direct the research. This chapter examines these questions by reviewing work on prior knowledge from the statistical literature and considering how these ideas relate to psychological categorisation. Obviously, this means that a thorough definition of prior knowledge is not possible in the introduction, but, as a rough guide, this review will be about processes which are useful to a categorisation system but are not considered by extant psychological models.

A further aim of this review is to introduce to the psychological community methods of modelling prior knowledge taken from statistics. This means that certain ideas of what constitutes prior knowledge are not covered here, such as Murphy and Medin's (1985) 'theories' or psychological essentialism (Medin & Ortony, 1989). Omitting these does not indicate any lack of assumed psychological importance; only that current statistical models tend not to utilise these ideas.

In the first section of this chapter, the type of categorisation model under consideration is described formally. The second section describes the bias / variance distinction (Geman *et al.*, 1992), a useful framework for examining prior knowledge. Section 2.3 reviews the approaches to incorporating prior knowledge into categorisation models, while Section 2.4 summarises the work described.

2.1 General Modelling Framework

In this section, the framework and notation which will be used to describe many of aspects of prior knowledge research is presented. In general, categorisation is taken to be a form of probability density estimation. Bishop (1995) and Ripley (1996) are excellent computational references for the ideas presented here, and Ashby and Maddox (1993) and Rosseel (1998) take a more psychological perspective. First, Bayes' theorem is described, then a successful model of psychological categorisation known as the General Context Model (Nosofsky, 1986) is outlined.

2.1.1 Bayes' Theorem

The basic aim of any classification system is to place an object into the correct category, given a set of previously classified similar objects and a quantity of information about the object to be classified. In other words, we would like to know the probability that the new item belongs to each of the possible categories given the information about the unseen object and the stored objects. Bayes' theorem provides us with a way of doing this. Before expanding on this however, we need to introduce some notation.

We will assume that all objects can be described by a set of dimensions, $x_1,...,x_d$, and that the input values from a single object can be grouped together to form a vector $\mathbf{x} = (x_1,...,x_d)^T$. There are $C_1,...,C_c$ mutually exclusive classes that an object could be placed into and let $\chi_k = \{x^n; n = 1,...,N_k\}$ be the set of stored category C_k exemplars. These assumptions are common to many categorisation models (e.g. Ashby & Gott, 1988, Medin & Schaffer, 1978; Nosofsky, 1986; Posner & Keele, 1968, 1970) and do not restrict the discussion significantly. Having described the situation using the above notation, we can write Bayes' theorem in the form

$$P(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k)P(C_k)}{p(\mathbf{x})} \qquad (1)$$

The term on the left hand side of Equation 1 is known as the *posterior* probability of the example being from class C_k, since it gives the probability of

the object belonging to the class after we have taken measurements of the object. From the right hand side, $p(\mathbf{x} \mid C_k)$ is the probability of drawing an object with $\mathbf{x}$ values from class C_k and $P(C_k)$ refers to the *prior* probability, that is the fraction of examples in class C_k, in the limit of an infinite number of observations. If we had to classify the new object without taking any measurements from the object, our best guess would be based on the priors. Finally, $p(\mathbf{x})$ is the unconditional density of $\mathbf{x}$ and is given by

$$p(\mathbf{x}) = \sum_{k=1}^{c} p(\mathbf{x} \mid C_k) P(C_k) \tag{2}$$

thus ensuring that the posterior probabilities sum to one. Having obtained the posterior probabilities from Equation 1, we minimize the probability of misclassifying the new stimulus by assigning it to the category with the highest probability (Duda & Hart, 1973). Of course, the denominator need not be calculated for comparison between classes since it is not class dependent.

The advantage in describing categorisation in this way is that many categorisation theories correspond in some way to Bayes' theorem. Regardless of whether a neural network is chosen as the implementation (e.g. Shanks, 1991), or 'feature sets' used (e.g. Tversky, 1977), or Kolmogorov complexity (Hahn & Chater, 1997), or even 'classical rules' (e.g. Allen & Brooks, 1991), the end result is the same: some form of posterior probability is calculated and a decision is made based on that.

2.1.2 The GCM

Nosofsky's (1986) General Context Model (GCM) is a popular model in the field of psychological categorisation (for examples of its application, see Ashby & Lee, 1991; Ashby & Madox, 1993; Lamberts, 1994; Lamberts & Shapiro, in press; Nosofsky, 1986, 1988a, 1988b, 1997) which fits neatly into general statistical ideas of probability density estimation. Together with its connectionist implementation, Kruschke's (1992) Attention Learning COVEring Map (ALCOVE), this model accounts for many of the traditional findings in categorisation (see Estes, 1994, for a review). As such, it will be used as examples of the type of categorisation model being considered when discussing how knowledge might be incorporated (Section 2.3).

In the GCM, the training examples are represented on a multi-dimensional space and assumed to be stored in memory, together with their category labels. When a test item (the probe) is presented, the similarity of that item to each of the possible categories is computed. The probe is then assigned to the category with the highest overall similarity.

More formally, the similarity of a probe, $\mathbf{x}$, to a stored exemplar, $\mathbf{x}^n$, can be seen as the probability that the probe was 'generated' from a particular distribution. This is usually taken to be either multivariate normal or Laplacian (exponentially shaped), depending on experimental setup (for a discussion on when to use each distribution, see Ennis, 1988; Nosofsky, 1988c; Shepard, 1988). Here, the normal distribution is adopted because of the ease with which comparisons can

be made with other statistical classification models (see Bishop, 1995, for a review). A measure of the similarity of the probe to the exemplar is therefore given by the probability of the probe being generated from the exemplar:

$$p(\mathbf{x}) = N(\mathbf{x};\mu^n;\Sigma) \tag{3}$$

where the mean of the distribution, μ^n , is given by the coordinates of the stored exemplar, $\mathbf{x}^n$. The covariance matrix, $\mathbf{\Sigma}$, is a d by d diagonal matrix with the elements σ_i^2 corresponding to the width, or variance, of the distribution for each dimension. When using a Euclidean distance measure, these are given by

$$\sigma_i^2 = \frac{1}{2c^2 w_i} \tag{4}$$

Nosofsky describes the weights w_i as *attention weights.* These weights are assumed to correspond to the degree of importance that participants attach to a given dimension during the learning process. For example, a high weight 'stretches' the dimension and means that distances are relatively larger (and therefore more important). The c parameter is called the specificity parameter, and controls the extent to which individual exemplars are distinguishable in memory. If c is high, the probability of a probe being generated from an a stored exemplar is relatively low, therefore it is less similar to the stored exemplars than if a low c parameter is used. This parameter can be thought of as a general smoothing parameter and is dealt with in more detail in Section 2.2.

To perform the classification task, the numerator in Equation 1 needs to be estimated. In the GCM, the first term can be expressed as

$$p(\mathbf{x} \mid C_k) = \sum_{n=1}^{N_k} P(\mathbf{x}^n \mid C_k) N(\mathbf{x}; \mathbf{x}_k^n, \Sigma)$$

$$= \frac{1}{N_k (2\pi)^{d/2} |\Sigma|^{1/2}} \sum_{n=1}^{N_k} \exp\left\{ -\frac{1}{2} \left(\mathbf{x} - \mathbf{x}^n\right)^T \Sigma^{-1} \left(\mathbf{x} - \mathbf{x}^n\right) \right\} \tag{5}$$

where N_k refers to all stored exemplars in category k. In other words, the probability of generating $\mathbf{x}$ from category k is determined by the summed probability of generating the probe from each of the stored exemplars, each probability weighted by the likelihood of generating the stored exemplar in the first place (the weight being simply $1/N_k$ for all exemplars). Finally, the participant is usually given an equal number of examples from each category, so that the prior probabilities, $P(C_k)$, are equal and can therefore be omitted from Equation 1 (although a frequency sensitive GCM has been developed for use if needed, by Nosofsky, 1992).

To summarise, categorisation in the GCM is assumed to be the process of estimating the posterior probabilities for each class, via a non-parametric[1] form of probability density estimation. The attention weights and sensitivity parameters are optimised on the basis of participants' responses, while all other parameters are specified through the experimental design. After applying

Equation 1, the resulting posterior probabilities are taken as the predicted proportion of responses for each category across participants.

By describing the GCM in this way, various other models of psychological categorisation can be considered in the same framework (such as Ashby & Townsend's, 1986, extension of General Recognition Theory (GRT); or prototype models by, for example, Homa, 1984; Posner & Keele, 1968, 1970; Rosch & Mervis, 1975). For instance, the density function of the GRT approach assumes only one distribution per category, rather than the mixture of N that the GCM does, but with a slightly more flexible covariance matrix. Similarly, the evidence that participants are sensitive to the correlation between features within a category (Anderson & Finchman, 1996) can be modelled by allowing the covariance matrix to be non-diagonal in either the GRT or GCM representations. The useful aspect of this generality is that the prior knowledge described below can also be seen as 'statistical' or model free.

The GCM is a model of generalisation. It is not intended to describe the learning process, that is the optimisation of the attention weights by the participants. These aspects of categorisation are modelled by ALCOVE (Kruschke, 1992), which uses the GCM as its representational base and gradient descent on the error to optimise the weights. ALCOVE is again a very successful learning model and was shown to replicate many of the standard findings on learning in categorisation, including the learning order in the six problems of Shepard, Hovland and Jenkins (1961), base-rate neglect (Gluck & Bower, 1988), and an

---

[1] 'Non-parametric' means that the classification model can approximate any shape of decision

appropriately low level of catastrophic interference (unlike standard back propagation networks: McCloskey & Cohen, 1989). Although the precise optimisation details are not relevant here, it is useful to note that ALCOVE is typical in its approach of equating learning in biological systems with minimising the discrepancy between the output of the model and the target values, that is, the training error.

boundary in the limit, as opposed to a 'parametric' model which can only take on certain forms.

2.2. The Bias / Variance dilemma

One caveat which needs to be attached to ALCOVE's learning mechanism (and that of many other models) is that the goal of any statistical model is not to maximise performance on the training data, but to capture the process that generated the data. The two are rarely the same, because noise invariably contaminates the input and output values. Although this problem appears to have very little to do with prior knowledge, Geman, Bienenstock and Doursat (1992) demonstrate that prior knowledge is central to the idea of maximising generalisation performance. Moreover, their analysis defines the fundamental problem facing any learning system, that of how much to pay attention to the data, and how much to rely on known information.

Geman *et al.*'s (1992) analysis involves the smoothing parameter in a model. In the GCM's case, this is the value of the c parameter (see Equation 4), illustrated in the four plots of Figure 2.1. These plots display outputs from four GCM's with different parameter values. The axes correspond to two input dimensions and the circles and crosses are training data given to the models. Note that in all four plots, the training data are the same. The solid lines are the models' decision boundaries, so that exemplars which fall above it are classed are crosses, while those that fall below are from the 'circle' category. The training data was generated from the function $y = \sin(2\pi x)$ (plotted as the dotted line), with some random noise added. What differentiates the models is the value of the c parameter. In the first plot, where c is high, the decision boundary is very jagged and seems to require smoothing out. Although its training error is zero, it

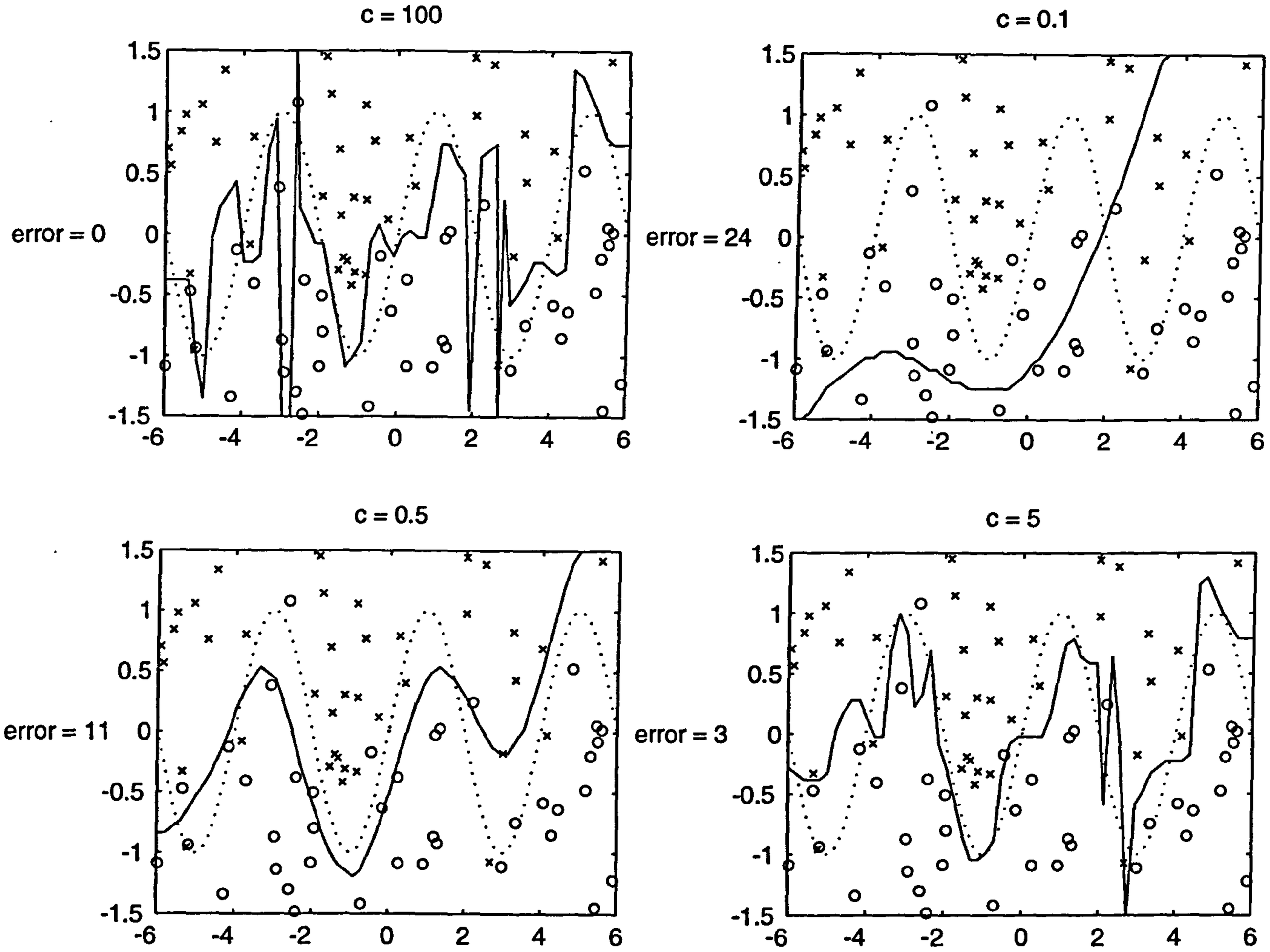


Figure 2.1 Four GCM models run with different c parameter values. Training error is shown to the left of each mode

doesn't seem to capture the underlying generating function. The second plot however, goes in the opposite direction. With the very low *c* value, the boundary is over-smoothed and is not flexible enough; consequently training error is high. The third and fourth models capture the sine curve much more accurately with *c* parameters between the two extremes of the first two models. Note that the training error does not appear to be a good predictor of the best decision boundary.

The *c* parameter controls the flexibility of the GCM. When *c* is high, the range of allowable decision boundaries is large (as shown in the first plot of Figure 2.1) and there is little risk that the model is not capable of representing the 'true' boundary, or little risk of bias. This flexibility comes at a price however, in that if the data turns out to be noisy, the decision boundary mirrors the noise. This jagged boundary could be smoothed out with a lower *c* value, but by doing so the range of allowable functions is restricted and bias may occur. It is worth emphasising that all models have their equivalents of the *c* parameter, which is generally termed the smoothing parameter. For instance, the number of hidden units in a neural network, the number of bins in a histogram, or the order of a regression polynomial, all have the property that they control the flexibility of the learning system.

Geman *et al.* statistically analysed these ideas by decomposing the generalisation error (which should be as low as possible) into a "bias" component and a "variance" component, as follows. The regression problem (or decision boundary formation if the problem is classification) is to construct a function $f(x)$

based on a set of training data, $D = \{(x_1, y_1), ..., (x_N, y_N)\}$, which approximates y, the function responsible for generating the data. Because the estimated function depends on the particular data set, it will be written as $f(x; D)$. A natural measure of the effectiveness of f as a predictor of y is the squared distance between them:

$$(f(x;D) - E[y \mid x])^2 \qquad (6)$$

Note that (6) above is the error at a *single* x-point (later on, we will integrate over all x values) and that $E[y \mid x]$ is used to emphasise that the 'true' value of y is needed, not just a sampled value. Because we are interested in what happens over all data sets, the expectation with respect to the data set needs to taken, that is, the average over the ensemble of possible D (for a fixed sample size N):

$$E_D\left[(f(x;D) - E[y \mid x])^2\right] \qquad (7)$$

There are two factors which might lead the error term in 7 to be large. First, it might be the case that $f(x; D)$ varies substantially with the individual data sets. For instance, with the high c value in Plot 1 of Figure 2.1, the estimated function would capture all the extraneous variability of the training sets. This contribution to the error term is known as "variance". Secondly, the estimated function may be far from the true function, averaged across the different data sets. In Plot 2 of Figure 2.1, although $f(x; D)$ would not vary over different samples, the average decision boundary would not capture the complexity of the generating sine curve, hence it is "biased". Geman *et al.* (1992) show that 7 can

be broken down into these two contributions to the error:

$$
\begin{aligned}
& E_D\left[\left(f(x;D)-E[y\mid x]\right)^2\right] \\
&= \left(E_D[f(x;D)]-E[y\mid x]\right)^2 && \text{“bias”} \\
&+E_D\left[\left(f(x;D)-E_D[f(x;D)]\right)^2\right] && \text{“variance”}
\end{aligned}
\tag{8}
$$

Thus, bias is defined as the deviation of the average estimated function from y, and variance as the expected variation of individual estimators from the generating function.

From the above discussion, it appears that there is a trade-off between bias and variance. In order to reduce the possibility of bias, the variance contribution must increase, while reducing variance entails an increase in bias. Put another way, placing too much belief in one's background knowledge risks missing the true nature of the category, but attempts to reduce this possibility lead to sensitivity to the idiosyncrasies of the particular data set we are given.

2.2.1 Simulations

To illustrate the workings of bias and variance, several simulations were carried out. These are adaptations of those documented by Geman *et al.* (1992). Two simulations are described, one involving a regression problem with small amounts of noise, and the other involving the same problem but with more added noise. Both use a radial basis regression network with Gaussian basis functions and optimum weights calculated using the pseudo-inverse technique (see Bishop,

1995; or Chapter 3 for more details). Here, the number of basis functions determines the smoothness of the solution (analogous to the *c* parameter in the GCM), so that a large number of functions allows a flexible regression curve.

The problem is a regression task, on 1 dimension. The input, *x*, is drawn from the line [0, 1]. Each example is given a target output of $y = 0.5 + 0.4\sin(2\pi x)$. Normally distributed noise with mean zero is then added to the target values, with a variance of 0.05 in the low noise condition, and 0.3 in the high noise condition. The training set, *D*, consists of 15 examples with their corresponding target values. Figure 2.2 shows an example data set from the high noise condition with the generating function shown by the dashed line.

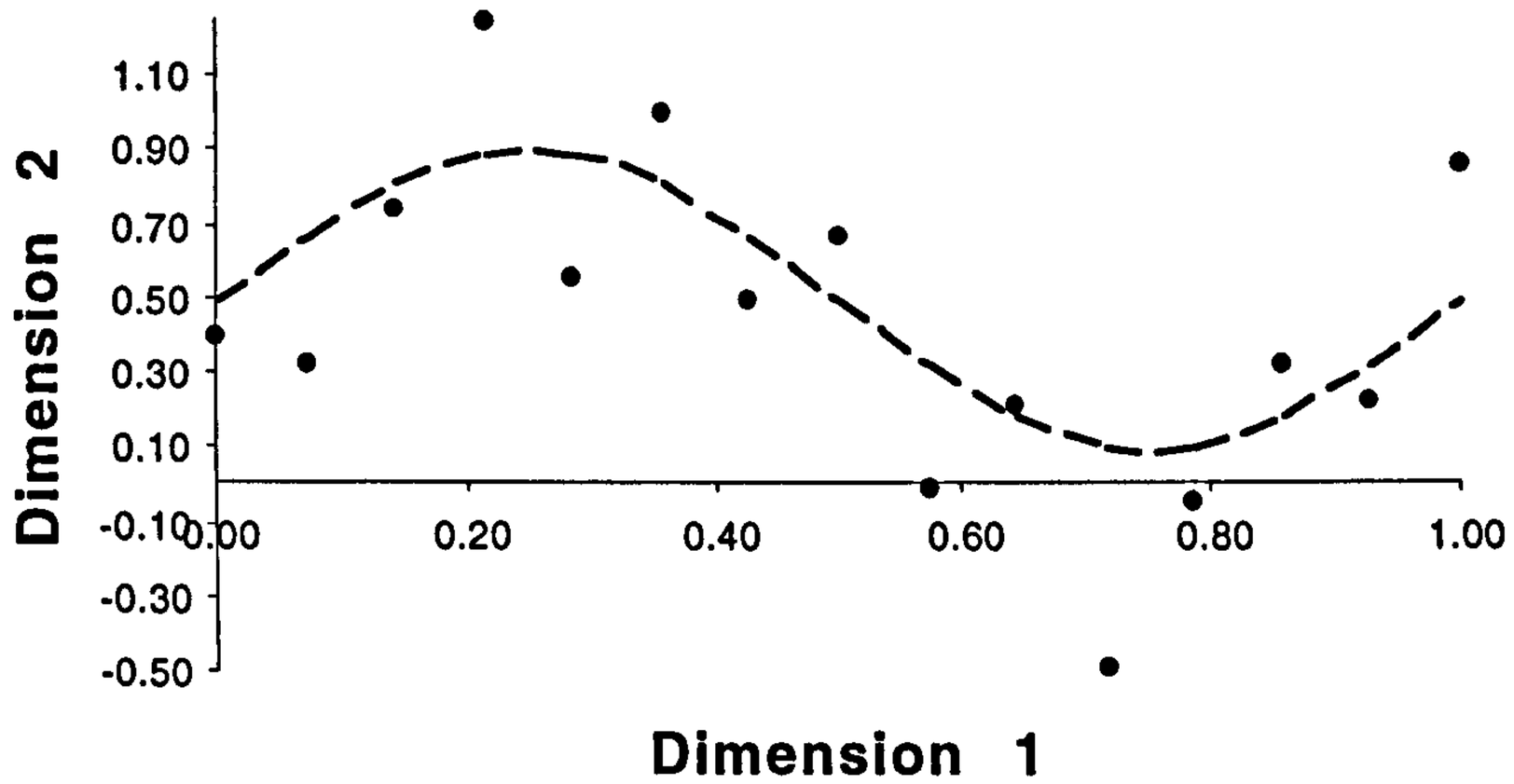


Figure 2.2 Example data set from the high-noise condition.

In each experiment, bias, variance and mean-squared error were estimated at different numbers of hidden units. This was done as follows. Fifty training sets were drawn, D^1, D^2,...,D^{50} and their corresponding estimators, $f(x, D^1)$, ..., $f(x, D^{50})$, were derived from the radial basis function (RBF) network. Denote $\bar{f}(x)$

as the average response at x: over all estimators, and bias and variance can therefore be estimated via the formulas:

$$\text{Bias}\,(x) \approx \left(\bar{f}(x) - E[y \mid x]\right)^2 \tag{9}$$

$$\text{Variance}(x) \approx \frac{1}{50}\sum_{k=1}^{50}\left[f(x;D^k - \bar{f}(x)\right]^2 \tag{10}$$

and the sum, bias + variance, is

$$\text{MSE}(x) \approx \frac{1}{50}\sum_{k=1}^{50}\left(f(x;D^k) - E[y \mid x]\right)^2 \tag{11}$$

Note that in this situation, $E[y \mid x]$ is known from the definition of the problem described above. Finally, overall bias, variance and MSE are found by integrating numerically over the range [0,1].

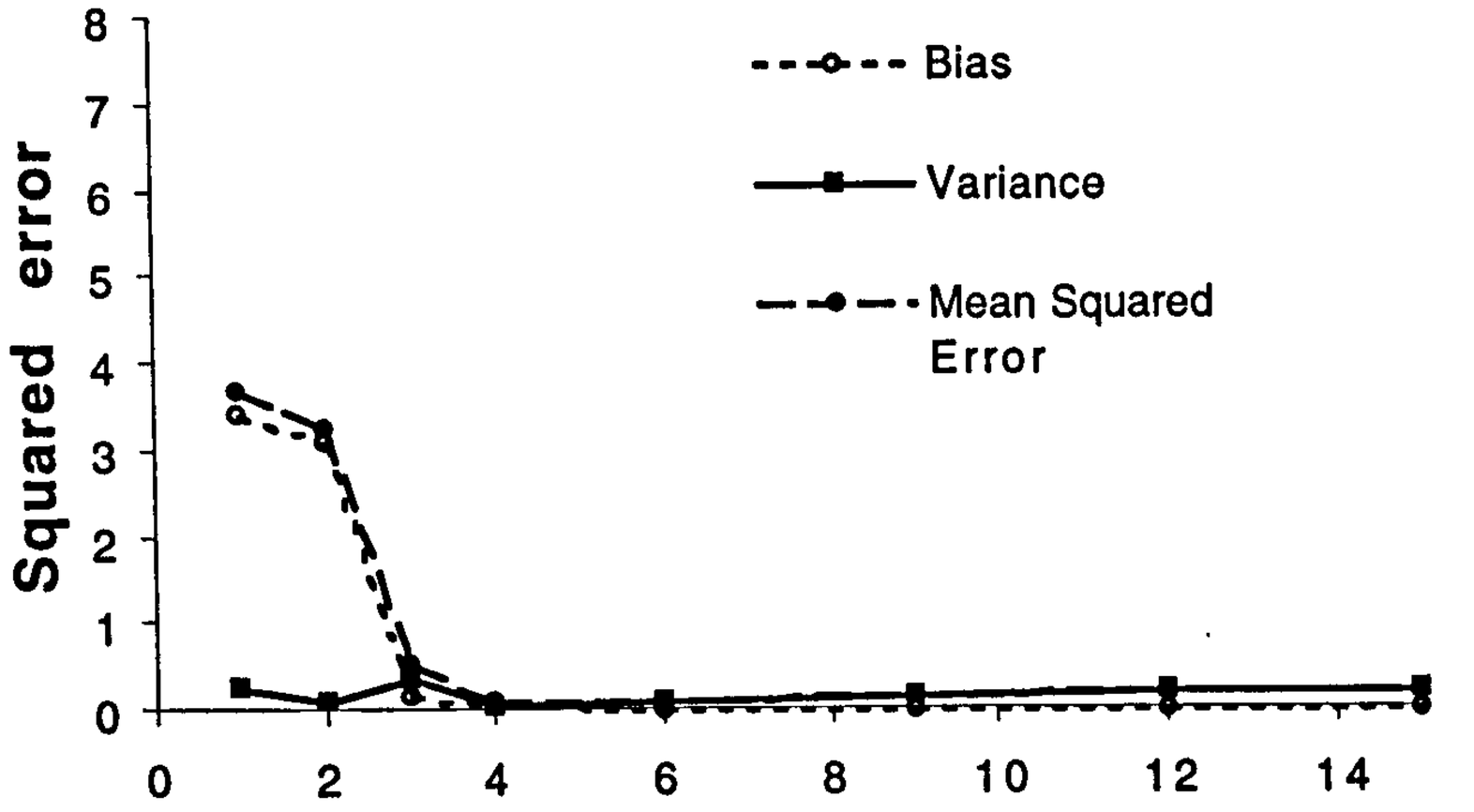
The results of the simulations are shown in Figure 2.3. The horizontal axes indicates the number of hidden units and the vertical axes the error value. The three lines on each panel refer to the integrated bias, variance and MSE respectively. In the top panel, with very little noise, bias falls sharply and is at minimum (zero) from 6 hidden units onwards. Initially, the network is too restricted to represent the problem, but 6 hidden units seems adequate to eliminate all traces of bias. Variance isn't a problem in this more deterministic task, although it does seem rise very slightly towards the end. The minimum mean squared error occurs just before this rise in variance, that is, at about 6

hidden units. A different story emerges in the second panel. Here, bias has again fallen to zero by 4 hidden units, but the variance component rises dramatically and is the principle contributor to the overall error. The extra noise added has meant that there is now a need to control the variance. Consequently, the ideal number of hidden units has fallen to 3 or 4 hidden units.

The simulations and analysis of the bias / variance distinction illustrate several important points for prior knowledge research. First, that with a finite number of examples, there is an optimum model complexity beyond which generalisation cannot be improved: different complexity levels either lead to higher bias or higher variance. For instance, in the high noise task above, 4 hidden units is the optimum level of model complexity and no method of choosing the level of smoothing can achieve better generalisation, whether Kolmogorov complexity (Schmiduber, 1997), Akaike's (1974) information criteria or validation sets are used. The question that Geman *et al.* (1992) then asked was, is this level of generalisation good enough? Or, more specifically, is it possible to achieve good generalisation with an environmentally appropriate number of examples and such empirically-based algorithms? According to Geman *et al.*, the answer is no – there just aren't enough examples to solve interesting problems. They argue that the only way generalisation will reach a sufficient level is to build "good" biases[2] into the model. By incorporating some prior knowledge, the range of allowable

[2] There is a slight problem with terminology here. The definition of *bias* in Equation 8 made clear that it was a contribution to generalisation error, and therefore can never be *good* (hence the use of quotes both here and in the original paper). Further, Geman *et al.* use the word *bias* to mean both the equation defined bias, and to mean inductive bias (from more traditional AI, which can be good or bad). To rectify this, *bias* will become 'error bias' or 'inductive bias' respectively.

functions is reduced, thus reducing variance. Bias is not increased however, because only incorrect functions have been eliminated.



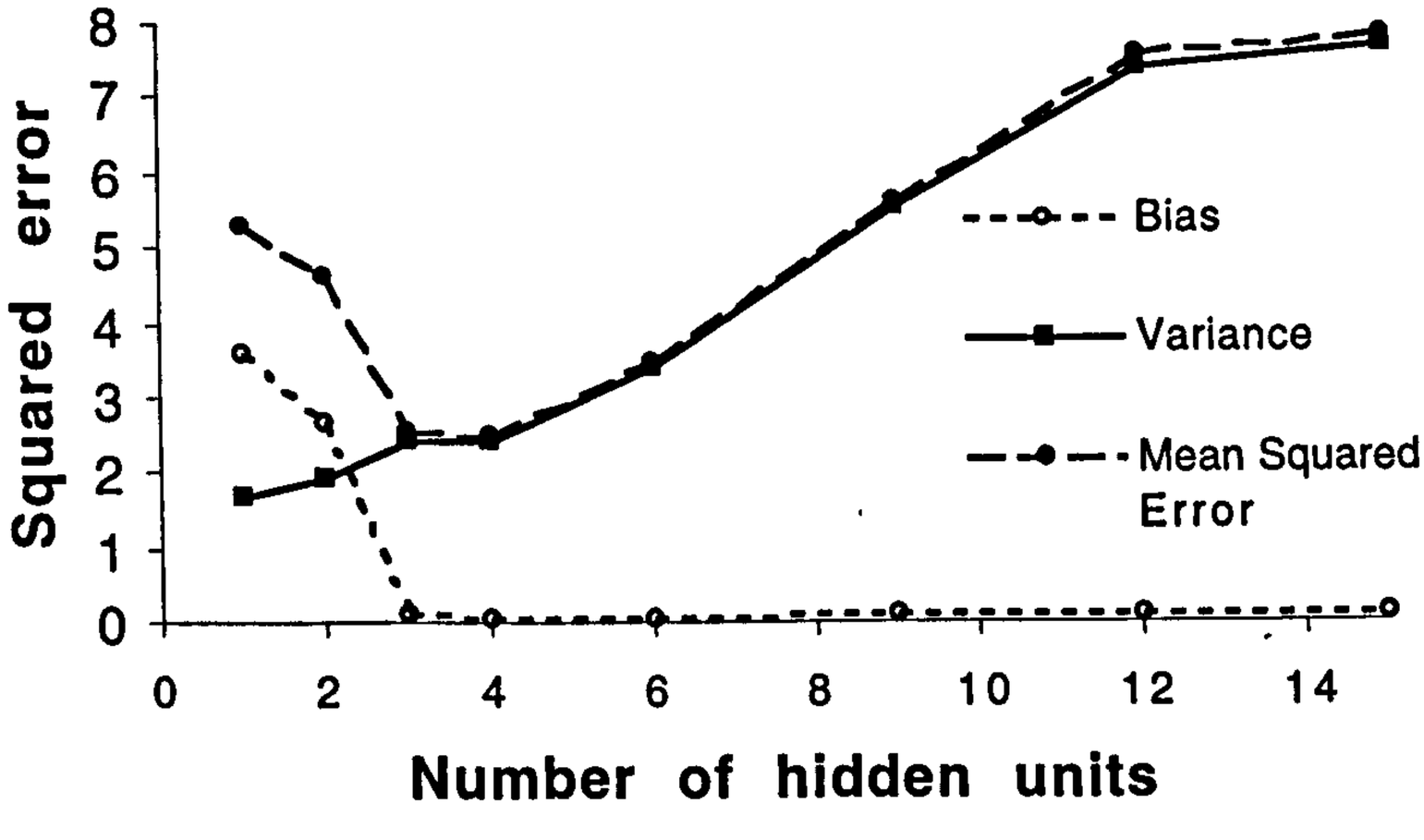


Figure 2.3 Estimates of the bias, variance and mean squared error for the High Noise and Low Noise tasks respectively.

Another way of thinking about the bias / variance dilemma is to examine extrapolation. To see this, look again at the four plots of Figure 2.1, and consider what the likely extrapolation behaviour of the models would be. The answer is that the regression curve would peter off into a straight line. Given the up and down nature of curve in the training area (and indeed the knowledge that a sine curve generated the data), this would seem an unlikely decision boundary. In fact, the only way non-parametric models like the GCM are going to be able make sensible predictions in this situation, is if more examples are found to cover the (infinite) space, hence the arguments in favour of prior knowledge put forward by Geman *et al.* (1992). Indeed, one reason humans may need 'rules', as opposed to a just an 'associative' mechanism, is to cover these extrapolation regions (see Erickson & Kruschke, 1998, for an experimental demonstration of the same point).

It is important to realise that this idea of increasing inductive bias in the model is distinct to setting the smoothing parameter. Reducing the level of smoothing certainly restricts the representational capacity of the model, *but at the risk of increased error bias.* Conversely, good inductive biases don't restrict the model indiscriminately; they only eliminate incorrect hypotheses. Of course, prior knowledge can help in setting the level of smoothing, but it is a conceptually different use of knowledge than that suggested by Geman *et al.* (1992).

The bias / variance analysis not only suggests that prior knowledge should be studied, it provides a very useful framework with which to examine knowledge. At least some aspects of prior information can now be seen as a way of

improving generalisation, or reducing the number of examples required to learn a concept.

2.3. Methods of Inserting Knowledge into Models

This section describes ways in which current models can be augmented with knowledge. The exposition follows a distinction made by Abu-Mostafa (1989, 1995), on the difference between knowledge chosen for its *information* value versus that chosen for its *complexity* value. The former is aimed at improving generalisation performance and is the type discussed in the previous section on Geman *et al.* (1992) work. On the other hand, complexity knowledge reduces the amount of computation needed to find the right hypothesis. For an example of complexity knowledge, consider an algorithm which adjusts a parameter in increments until some level of error is reached. Knowledge of the approximate value of this parameter will reduce the number of iterations required by this algorithm to reach the desired level of error. Hence, the computational requirements are reduced. Notice that the theoretical generalisation performance is not necessarily improved – the algorithm may well have achieved the same level of performance without this hint.

Another difference between the two types of knowledge is the extent to which they can be considered algorithm specific. The usefulness of knowledge which reduces computational demands must necessarily be dependant on the type of algorithm involved, because different algorithms solve a problem in different ways. Getting an idea of the parameter value, as in the example above, is far less useful for an algorithm which solves the problem analytically than it is for an incremental one. Conversely, knowledge used for its information content tends

to be useful for all algorithms: specifying that the target function is linear gives the model more information to work with, whatever the method it uses.

Further insight can be gained by examining the link between knowledge given to an algorithm and training examples. When the bias / variance distinction was discussed above, it was made clear that generalisation could only be improved by providing more training examples, or by incorporating prior knowledge of the target concept. In other words, the generalisation improvement from some prior knowledge can be seen as equivalent to a certain quantity of extra training instances given to the algorithm. (Heit, 1994, 1995, makes a similar point). Abu-Mostafa (1995) has formalised this intuition using a theoretical AI tool known as the VC Dimension (Vapnik & Chervonenkis, 1971; and see Haussler, 1984, for a useful introduction) and demonstrated how different types of prior knowledge can be quantified. In terms of the information / complexity issue, only knowledge used for its information content is appropriately thought of as equivalent to extra training examples.

There is some overlap between information value and complexity, however. As Abu-Mostafa (1995) puts it, "without sufficient information, no algorithm, slow or fast, can produce a good hypothesis. However, sufficient information is of little use if the computational task of producing a good hypothesis is intractable". Further, the overlap becomes greater within psychology because of the problem of experimentally distinguishing between multiple presentations of the same examples (a complexity issue) versus presentation of different examples (an information issue) (see Barsalou, Huttenlocher, & Lamberts 1998, for a

demonstration of the difficulties), an issue which will be returned to in the conclusion. None-the-less, some situations are clear cut, and the information / complexity dichotomy is useful for breaking down some psychological approaches and drawing links between different disciplines.

2.3.1 Complexity

One of the most obvious ways of incorporating knowledge is to set the initial weights to some value which is believed to benefit learning, rather than to a random value. For example, Kruschke (1992) suggested that dimensions which are particularly salient to a participant before a categorisation task can be modelled by assuming a higher weight on the appropriate dimension (Vandierendonck & Rosseel, 2000, carried out experiments to empirically confirm this). So, what change does this prior knowledge lead to in the learning process? In general, the answer is that it reduces the number of iterations required before the solution is reached. Asymptotic generalisation performance, as discussed in the section above, is not improved directly[3]. The same weight solution is reached as without the knowledge, but more quickly. Of course, improving learning speed may be exactly what is desired in some situations. Moreover, the psychological experiments examining prior knowledge are rarely concerned with asymptotic performance - differences are usually observed at the beginning of learning. At this point generalisation performances do differ, which

[3]It should be noted however, that setting the weights can lead to different minima being discovered, so generalisation performances may be indirectly altered.

has lead a large number of psychological researchers to use initial weight settings to model the prior knowledge results, as discussed below.

Giles and Omlin (1993) carried out an empirical investigation of the effects of incorporating planned initial weights into a network which helps to illustrate the advantages and disadvantages of this method. The network itself was a recurrent network with feedback weights (nonadjustable) from the output to the input units. The task was to learn a deterministic finite-state automaton (DFA) from a set of positive and negative example strings. In the hidden layer, the network was provided with a sufficient number of units so that it could represent the particular DFA under consideration. In other words, there was at least one unit for each DFA state. Knowledge about which state transitions were correct was incorporated by setting the weights between a pair of units to positive values (a valid transition) or negative values (not a valid transition). The precise magnitude of the weight 'hint' (H) was varied between H=1 and H=7. Finally, Giles and Omlin also varied the veracity of the knowledge from 'correct' (all transition rules inserted) to 'incorrect' (some rules inserted) to 'malicious' (random DFA's inserted).

The results of the study were as follows. First and most obviously, Giles and Omlin (1993) found that if the knowledge was correct, the number of epochs required to reach criteria fell monotonically with hint strength. This knowledge moved the network closer in weight space to a good minima, and so fewer weight updates were required. Secondly, when partial solutions were injected, training time decreased up to a H=6, but then began to increase. Too high a level

of knowledge apparently pushes the algorithm into local minima which it has difficulty escaping from. For the malicious rules with high strength, criteria was not reached, but at low levels training time was reduced over a network having no pre-programmed weights. Presumably this was because some of the randomly generated DFA's (malicious knowledge) contained some transitions which were correct, and therefore benefited the system. In summary, although the final solutions obtained were very similar to the network which had no weights pre-programmed, the training time varied considerably with the experimental manipulations. Thus, it could be concluded that the effect of initial weight programming gradually gets reduced as learning progresses. A further point to be made is that the network can wipe out inappropriate knowledge (Omlin & Giles, 1996), or indeed veridical knowledge, if it disagrees with the data.

A good psychological example of the use of initial weight programming is provided by Choi, McDaniel and Busemeyer (1993). Their aim was discover whether extant formal categorisation models were capable of fitting human rule-learning behaviour (e.g. Salatas & Bourne, 1974). The data from rule-learning experiments indicates that there are clear differences in the ease with which humans learn logical rules, with conjunctive the easiest and biconditional the most difficult on a two dimensional, in- or not-in classification problem. The task for the modellers was to see whether these biases might be built in to the categorisation models. Several models were tested, among them ALCOVE. Biases were introduced by setting the weights connecting the hidden units (exemplar nodes) to the output units (category nodes) to positive or negative

values. For instance, to incorporate the bias that people have for assigning the {1,1} item to the positive category, an excitatory weight was placed on the connection from the {1,1} hidden node to the positive output node. ALCOVE succeeded in reproducing the order of difficulty of the classification rules but, interestingly, only two out of the four biases suggested by Bourne's (1974) rule learning model needed to be explicitly included. The remaining biases emerged as a consequence of the interaction between the other two and ALCOVE's learning algorithm.

The psychological research on function learning has also been concerned with how easily participants solve some problems over others, and how to model this. This area (e.g. Naylor & Clark, 1968; Brehmer, 1974; Koh & Meyer, 1991; Busemeyer, Byun, Delosh, & McDaniel, 1997) examines how people learn mappings from continuous input to continuous output dimensions, such as that from visual to proprioceptive dimensions (Bedford, 1989). Several biases in the order in which people learn different functional forms have been discovered, for example that increasing functions are learnt faster than decreasing functions (Naylor & Clark, 1968). Busemeyer *et al.* modelled these biases by using a 'proportional prior knowledge' assumption to programme the initial weights. In the linear case, this meant that the minimum observed input value is mapped onto the minimum observed output value, the maximum input is mapped to the maximum output, and intermediate stimuli are mapped proportionally. This example illustrates a potential problem with using weight pre-programming – some way is needed of calculating which weights to programme and to what

degree. In some situations, this task might be more computationally demanding than learning the problem from scratch.

This section has discussed an example of initial weight programming from the engineering literature and several psychological examples. Although the two domains have used the same method of building in knowledge, they seem to differ in what they expect from this knowledge. In the engineering literature, researchers have either focused on improving learning speed (e.g. Giles & Omlin, 1993; Towell *et al.*, 1990) or avoiding local minima (Suddarth & Holden, 1991), but psychologists have focused on reproducing order of acquisition effects. This difference in rational begs the question of whether psychologists should be using this method as a general approach to knowledge representation, given that it is not really tied to a statistical rationale. On the one hand, as described above, the method does a good job of capturing the data. On the other, there are some theoretical problems with the technique. First, initial weight programming is a very temporary form of knowledge. Examples from the environment will wipe out the initial weights settings, meaning that any noise in the environment (which is usually absent from psychological experiments) will be reflected in the final solution. Evidence from various sources (Wisniewski & Medin, 1994; Wisniewski, 1995) indicates that prior knowledge has a persistent and interactive effect on learning, contrary to what initial weight models might predict.

Secondly, pre-programmed weights are only an advantage to some learning algorithms, namely gradient descent. If, for example, participants in a function

learning task are performing a more traditional hypothesis searching technique (as Delosh, 1999, provides evidence for), then initial weights are irrelevant. Similarly, if the problem is linear, then the appropriate weights can be obtained analytically, or if genetic algorithms are used as the optimisation procedure, then a completely different idea of implementing 'closeness' in solution space is required.

2.3.2 Information

In this section, methods of incorporating knowledge are grouped by the order in which they might occur in the processing of the network, that is, starting with how knowledge can be built into the inputs and moving 'up' towards the outputs. Although it could be argued that this manner of presentation confuses implementation issues with statistical ones, it was felt that specific models provide concrete examples of the statistical knowledge and improve the exposition. Moreover, both issues are discussed where possible. For example, when discussing inputs into the learning algorithm, a statistical rational is first presented, then some specific approaches to dimensionality reduction.

Dimensionality reduction

When faced with a learning situation, the organism must first decide which features of the object are going to play a part in the learning process. In terms of the GCM, the question is, which dimensions should the object be encoded on? An obvious answer is "All of them", and the learning algorithm can sort out

which ones are useful and which aren't. There is a problem with this blanket approach however, which is that there may be an insufficient number of examples to specify a mapping in a high dimensional space. This problem is known as the *curse of dimensionality* (Bellman, 1961) and indicates that some sort of knowledge is required to specify the appropriate dimensions before learning commences.

To illustrate this idea, consider a task where an algorithm must form a linear rule which classifies unknown examples into one of two categories. In the case where the objects are described on two, binary dimensions, there are two possible decision boundaries (separate based on the Dimension 1 value, or on Dimension 2). This means that 3 training examples are required to specify the rule in the worst case, assuming sampling without replacement. Now imagine the task on three dimensions. Here, we have three potential decision boundaries, and the number of examples required to specify a given hypothesis has risen to 4. This toy problem illustrates that increasing the dimensionality of the hypothesis space requires more examples to specify the mapping or, put another way, an excess of dimensions for a given number of examples leads to poor generalisation.

As a consequence of the curse of dimensionality, it is common to perform some kind of dimensionality reduction before presenting patterns to the network (Bishop, 1995). This can take the form of simply eliminating dimensions which appear correlated with other dimensions, to performing linear data compression processes such as principal components analysis or multi-dimensional scaling, or

even non-linear neural network methods. As Bishop remarks, the distinction between pre-processing and performing the regression begins to get blurred here, and the selection of features can basically be considered a form of unsupervised learning. Nevertheless, it is clear that performing the kind of learning procedure inherent in ALCOVE on the retinal output is impractical; some form of feature selection is required.

Of relevance here is work done on the interaction between conceptual learning and perception carried out by Goldstone, Schyns and colleagues (e.g. Goldstone, 1994; Goldstone, Steyvers, Spencer-Smith, & Kersten, 1999; Schyns & Rodet, 1997; Schyns, Goldstone, & Thibaut, 1998). The main thesis behind their work is that the features on which objects are described do not remain constant through a categorisation process, as the GCM would maintain, but that the act of classification learning feeds down to influence how the object is perceived. For example, Goldstone (1994) provides evidence that people develop sensitivity to regions of novel, face dimensions and that they carry this perception into other categorisation tasks. First, a two dimensional grid of faces was created by morphing two faces to form one dimension, and another two to form the second dimension. Then, one group of participants were taught a categorisation task involving a decision boundary on the first dimension, another group on the other dimension. In a subsequent categorisation task where the relevant and irrelevant dimensions were reversed, negative transfer effects were observed. This study demonstrates that people are quite able to form their own feature set based on the demands of the classification problem: from the retinal coding of the faces, some

kind of dimensional reduction takes place, which is then used as the input to a categorisation procedure.

At a more cognitive level, Markman (1990) describes evidence of children's constraints for acquiring concepts. This includes the whole-object assumption, whereby a novel category label is assumed to refer to the whole object, rather than its constituent parts. Without this constraint, a child has no way determining that the label "tree" refers not to the "branches", but the whole structure for instance. Another suggested example is the bias young children seem to have towards an object's shape (see Ward, 1993, for a review). When children are taught that an object is called a "Dax", for instance, they classify other new objects on the basis of shape, even if rejected objects have the same colour or texture. Although interesting enough in themselves, the important point to pick up from these studies is that children clearly have preconceived ideas about which dimensions are likely to be important for label learning.

Structural invariances and preprocessing

As described above, Geman *et al.* (1992) referred to the need to incorporate "good" biases for the problem at hand. These would eliminate incorrect hypotheses and therefore lower bias without increasing variance. Although much of this paper takes its motivation from this idea, Geman *et al.* suggest some particular low-level, perceptual knowledge, known as *invariances.* In general, invariances are said to occur when outputs of a classification task are known to be the same under some transformation of the input value. A good example is

the recognition of objects in two-dimensional images: their classification is unchanged whether the object is rotated, translated or linearly scaled (corresponding to moving closer or further away from the eye), despite the considerable change in input values that arises. Other examples are provided by Shepard (1989), such as colour constancy (Land, 1964), or any domain in which scale-invariance holds (Chater & Brown, 1999).

Barnard and Cassent (1991) identify three approaches to the implementation of invariances. First, it is possible to train the network the invariances by example. Any number of examples can be generated by performing transformations on known 'true' examples. This is a simple form of the 'hints' idea developed by Abu-Mostafa (1995) and is discussed in more detail in the *Error functions and hints* section below. A second approach is to perform some kind of pre-processing to extract features which are themselves invariant. Such features are often based on moments of the data. For instance, translation invariance can be achieved by extracting the deviation of the coordinates from the mean, and basing the classification on those values. Finally, knowledge can be incorporated directly into the structure of the network in a variety of ways, such as with *shared weights* (e.g. Rumelhart, Hinton, & Williams, 1986; Fukushima, 1988). Consider an example provided by Bishop (1995) on building in translation invariance to a classification network. Here, the network is hierarchical with a pixel-based input of an object image and the first hidden layer consists of nodes which respond to a local receptive field. However, instead of being fully connected, each weight within a field is constrained to be the same as the corresponding pixel weights making up the other receptive fields. This

means that if an object falls in one node's field, weights into it are updated not just for that node, but for all the nodes on that layer. In the second hidden layer, a set of fixed weights computes a simple average of the activation from the units in the first layer so that, wherever a given object falls in the entire field, the node receives the same amount of activation.

Goldstone *et al.* (1999) use a form of weight sharing to demonstrate how a network might perform object segmentation in a similar way to humans. They were modelling a task where the segmentation participants chose was influenced by previous category learning, subject to the constraints of the Gestalt laws of good continuation and closure. The model was a feed forward network with a pixel-based input, a hidden layer of 'feature detectors' and outputs corresponding to several categories. A competitive learning algorithm was used with a slight modification. This adjustment meant that detectors that were useful for categorising an input pattern now became more likely to win the competition to learn the pattern. As the model stood, it reproduced the basic finding that different feature detectors developed depending on the categorisation structure used in training. However, the pixels that the hidden units became specialised for were not grouped appropriately; nobody would decompose the object in to the highly distributed pattern that emerged. This was rectified by using topological constraints on the detector creation though weight sharing. To produce detectors that respond to cohesive, contiguous input regions, input-to-detector weights were now adjusted not just for the 'winner', but for close neighbours as well. Input-to-detector weights also spread to each other as function of their orientation similarity. This meant that detectors now followed

principles of good continuation, for example dividing an 'X' up into two crossing lines rather than two kissing 'V's, because the two halves of a diagonal line will be linked by their common orientation.

These examples illustrate how prior knowledge can be seen as a way of *constraining* a learning system: without the weight sharing in Goldstone *et al.*'s (1999) model, the network was capable of learning many more hidden representations. Although these other representations may well have achieved a lower training error, our constraints eliminate these possibilities in order to maximise generalisation behaviour. Or, put another way, our prior knowledge takes the place of the number of extra examples which would be required to reject these 'untrue' hidden representations.

Model order selection

In the simulations shown in Figure 2.3, the optimum number of hidden units were found to be 4 and 6 respectively. Clearly, in real problems setting the smoothing parameter by minimisation with respect to the true generalisation error is impossible, nor does the training error provide much help, as we saw in Figure 2.1. How then, could the smoothing value be determined, and how might knowledge help?

There are certainly theoretical approaches to the problem, such as Kolmogorov Complexity (Schmidhuber, 1997) or Akaike's information criteria (Akaike, 1974). However, these will not be discussed here because this section is

concerned with how to incorporate knowledge into computational models, which implies that the techniques need to be psychologically testable, and perhaps biologically plausible. It does not seem that psychological practice is sufficiently precise to distinguish between the different theoretical approaches yet, nor are the neurosciences sufficiently developed. However, there are more practical approaches which will be discussed.

Direct knowledge of smoothing

Specific knowledge about smoothing can be useful for an organism and is reasonably easy to incorporate in a model. For instance, knowledge that the domain is noisy should encourage a relatively large smoothing value. Advantage can also be gained by knowing the complexity of the problem. For example, if it is was known that a category was generated from a bi-modal Gaussian distribution, then the number of Gaussian basis functions in a mixture of experts model could simply be set at two. Similarly, in a network with binary hidden units, the knowledge that the true decision boundary formed a logical OR gate could be realised by using a single hidden unit. Of course, knowing the complexity of the problem is not the entire answer – being aware that the regression function is a 10^{th} order polynomial is of no use if only ten training examples are available – but it can provide upper bounds on the complexity needed in a given situation.

The psychological literature is surprisingly thin on this topic, given the prevalence of the GCM and other similarity-based models within categorisation.

What tends to happen when these models are fitted is that the smoothing parameter is optimised to provide the best fit to the psychological data (for example, Ashby & Lee, 1991; Nosofsky, 1986, McKinley & Nosofsky, 1995), thereby avoiding the questions about how estimation is carried out by humans. An exception to this is Lamberts (1994), who carried out a series of experiments demonstrating that people are capable of learning a set of items as individuals, then (in a second stage) generalising differently depending on the labelling of those examples. This was shown by fitting the GCM to the different sets of responses and showing that the c parameter was different in the two labelling conditions. One interpretation of this is that people initially learn everything they are presented with, and then estimate the smoothing parameter when they are required to generalise (and know more about the domain they are in). Assuming an infinite memory capacity, this is a highly intelligent strategy. If on the other hand, the learner is forced to make some kind of smoothing or abstraction during the learning process, the smoothing parameter is tied to the other weights and cannot simply be changed at the last minute. For instance, carrying out the learning task and then adding hidden units to a neural network will not produce good results - the entire learning process must take place again.

Dynamic architecture models

Several types of models have been developed which alter their representational capacity as a function of learning. These can be divided into two classes: growing networks, (Fahlman & Lebiere, 1991; Mareschal & Schultz, 1996;

Prechelt, 1997; Quartz & Sejnowski, 1997; Westermann, 2000), which expand their representation as learning takes place; and pruning networks (Hanson & Pratt, 1989; Mozer & Smolensky, 1989), which reduce their representational power during learning. Both types are a practical way of estimating the structure of the network, but there are differences in the philosophy and reasoning behind their approaches to the problem.

Growing and pruning algorithms change their representational capacity by respectively increasing or decreasing the number of hidden units. In constructivist nets, the architecture initially contains a small number of units, then adds one when the network is unable to reduce the training error past a given criteria. For instance, in Westmann's (2000) model, a new hidden unit is added when the error gradient is less than an *a priori* determined parameter value. In deconstructive networks, there is a large number of hidden units at the start, but those nodes which are considered irrelevant are removed from the network during learning

Constructive methods have several advantages over static models. First, they represent a practical way of estimating the number of hidden units for a particular task. Instead of having to invest large amounts of time in training many different-sized nets with to see which performs the best (using a validation set, say), the dynamic algorithm needs only one run – it will 'discover' the appropriate number. For example, a constructive algorithm offers an easy way of estimating that the 'noisy' problem in Section 2.2.1 needs only 6 hidden units. Secondly, in the case of constructive algorithms, there may be some advantage

for the optimisation algorithm (back-prop, quasi-newton etc.) in 'starting small' (Elman, 1993), or having some structured path through the weight space. This does not directly benefit generalisation, but, as discussed above, it may reduce training time or avoid local minima. Finally, they may achieve some generalisation advantage over the optimum performance of a static network (e.g. Westmann, 2000) because the final structure is slightly different to the standard feed-forward network in connectivity. This structural difference may reduce error bias as described above in Section 2.3.2, *Structural invariances and preprocessing.*

Both types of network have been used for psychological modelling. Schultz and colleagues (Mareschal & Shultz, 1993; Shultz & Schmidt, 1991; Mareschal & Shultz, 1996) have shown how constructivist networks model various Piagetian stage changes in child development, such as the seriation task (Piaget, 1965). The general idea is that the adjustment of weights corresponds to Piagetian 'assimilation', or quantitative changes in behaviour, while the addition of the hidden units correspond to 'accommodation', or qualitative changes. They've also been used to explain the U-shaped learning curves in English past-tense acquisition (Westmann, 2000) and personal pronouns (Schultz, Buckingham, & Oshima-Takane, 1994). On the biological side, pruning networks have been used to demonstrate the computational rational behind the apparent 'suicide' of cells in the developing nervous system (Brown, Hulme, Hyland, & Mitchell, 1994), and Quartz and Sejnowski (1997) provide a useful discussion of the biological importance of constructivism. Slightly surprisingly, dynamical models have not yet been used to model categorisation experiments. It would

seem likely that they might capture order of acquisition effects, such as the benchmark Shepard *et al.* (1969) problems, or indeed any result where a 'simple-first' strategy looks plausible.

Modular architectures

Instead of choosing one network with a single smoothing value, another approach is to use many networks with a range of smoothing values and average the output. This results in a modular network, or a committee of networks, which not only eliminates the need to select a single smoothing value, but may also improve generalisation beyond that which the best single network is capable of. Further, modular techniques provide a convenient way of combining different forms of knowledge in a system: different types of knowledge can be built into each module and the resulting decision is some weighted combination of all these sources. Because of this, and because later chapters describe models which are modular in nature, a more lengthy discussion is provided than other model selection techniques.

As described by Bishop (1995) and Perrone (1994), committees are a series of L networks joined together so that all take the same inputs. Each component has a different level of smoothing, and their outputs are combined so that the end result is just one decision. For example, the committee might consist of three GCM's with different c values and the overall output being the average decision from all

three. Now, assuming a squared error function, the average error made by the networks acting *individually* is given by

$$E_{AV} = \frac{1}{L}\sum_{i=1}^{L} X[e_i^2] \tag{12}$$

where L is the number of networks, X refers to the expectation and e_i to the error from each network as specified in Equation 6. A simple committee could be formed by taking the outputs of the individual modules and letting the output of the committee be the average of these networks. The committee prediction is therefore

$$y_{COM}(x) = \frac{1}{L}\sum_{i=1}^{L} y_i(x) \tag{13}$$

where y_i is the output from the networks individually. The average error for the committee becomes:

$$E_{COM} = X\left[\left(\frac{1}{L}\sum_{i=1}^{L} y_i(x) - h(x)\right)^2\right] = X\left[\left(\frac{1}{L}\sum_{i=1}^{L} e_i\right)^2\right] \tag{14}$$

If the assumption is now made that the errors e_i have zero mean and are uncorrelated, so that

$$\begin{aligned} &X[e_i] = 0, \\ &X[e_i, e_j] = 0, \end{aligned} \qquad \text{if } j \neq i \tag{15}$$

Combining Equations 12 and 14 then gives

$$E_{COM} = \frac{1}{L^2}\sum_{i=1}^{L} X[e_i^2] = \frac{1}{L}E_{AV} \tag{16}$$

Thus, simply averaging the output of these networks reduces the generalisation error by a factor of 1/L, due to a reduction in the variance component of the error. Some intuitive understanding of this rather startling result can be gained by considering the standard statistical practice of averaging data points to obtain a better estimate. For example, if we were required to find the level of depression in a population, the sensible approach would be to take the average score from many people, that is, form a *committee*. Of course, one could argue that a single, well-chosen person might be more suitable, but usually the reduction in variance from averaging is considered a better estimate. It can also be seen that there are times when the average estimate from the group is closer to the population score than any individual's, although it is not necessarily the case.

One problem with the committee described above is that the errors of the individual networks may well be correlated, for instance in a situation where all the modules are trained on the same data set. This correlation breaks the assumptions described below Equation 15, and consequently less of a reduction in error is achieved (although the committee will never have more error than the average of the individuals, as Bishop, 1995, proves). Because of this possibility, techniques have been developed for designing modular architectures which either reduce the correlations (for example, Jacobs, Jordan, Nowlan, & Hinton, 1991, and reviewed by Jacobs, 1995) or minimise the effect the of correlations (Perrone, 1994, Bishop, 1995). Of these, the mixture-of-experts (ME)

architecture developed by Jacobs and colleagues (Jacobs *et al.*, 1991; Jacobs, 1995; Jacobs, 1997) is particularly relevant here because of the published applications to perception and categorisation (for example Erickson & Kruschke, 1998).

A ME architecture aims at learning task-decomposition in the sense that it uses different networks to learn input-output training patterns from different regions of the input space. This is achieved by having a group of expert networks and a 'gating network'. The role of the gating network is to allocate different input patterns (or areas of the input space) to the different experts. The structure and complexity of the experts is arbitrary, but the gating network must have as many output units as there are experts and the activation of these output units must sum to one. The final output of the network is given by:

$$y = \sum g_i y_i \qquad (17)$$

where y_i denotes the output of the ith network and g_i is the respective weight. The output nodes of the gating network respond differently to different parts of the input space, which allows them to control the extent to which a given expert influences the overall output. Optimisation of a ME model proceeds as follows. Each expert receives training in proportion to their success at predicting the target value. This means that experts which do well for some training patterns continue to get better, and other networks receive less and less training. In this way, different modules become experts at different tasks, and their outputs become less correlated.

An appropriate example is provided by Erickson and Kruschke (1998), who used the ME architecture to explain the interaction between rules and exemplar learning in a categorisation task (for other modular categorisation models, see Ashby, Alfonso-Reese, Turken & Waldron, 1998; and Vandienendonck, 1995). The expert networks were ALCOVE, as described above, and a rule module, consisting of a small number of hidden units capable of learning rules like "an input belongs to Category A if it is a high value on dimension one". Note that the two experts are of differing complexity: ALCOVE is capable of a decision boundary of arbitrary complexity, while the rule module uses only linear decision bounds. In their experiments, Erickson and Kruschke demonstrated that in some areas of the input space participants used a rule, while in others they used ALCOVE. The ME model reproduced this finding and several other experiments, leaving Erickson and Kruschke to conclude that neither expert was a sufficient model on its own; the interaction of the two was necessary.

To conclude this section it is worth considering the generalisation properties of modular networks in a bit more detail. Let's say that once again we were faced with the problem in Section 2.2.1 and we had estimated 4 hidden units for the noisy problem, that is number which gave us the lowest generalisation error. Could we improve generalisation by adding another module, say a cyclic curve module, to the system? This makes the system more complex, in the sense that a greater range of solutions can now be found. However, it is not the case that we have moved further to right in Figure 2.3 (which would imply that generalisation could not be improved), but that we have moved to a different graph altogether -

the class of models we are using has now changed. The knowledge that a cyclic curve might be useful can improve generalisation, not through eliminating incorrect hypotheses, but by changing the algorithm.

Error functions and 'hints'

Models typically use an error function which is based on the deviation between the target function and the estimated output. The error function can in fact be altered to incorporate different forms of prior knowledge, methods of which are described in this section.

Regularisation theory involves adding a term to the error function to penalise some 'unwanted' aspect of the end mapping. The total error then becomes

$$E_T = E + \nu\Omega \tag{18}$$

where E was the old error term, Ω is the penalty function and ν a parameter controlling the extent to which the penalty term is weighted in the optimisation (Bishop, 1995). The regularisation term is used to control the complexity of the model (in the same way as the *c* parameter does in the GCM) and therefore provides a means of finding a suitable bias / variance compromise. Different forms of regularisation term are used for different problems. A common one however, is used to reduce the curvature of a regression function (or decision boundary). Referring again to Plot 1 of Figure 2.1, the reason this boundary seems intuitively implausible is that it is very jagged, that is, has a large amount of curvature. A term like

$$\Omega = \frac{1}{2}\int\left(\frac{d^2y}{dx^2}\right)^2 dx \qquad (19)$$

which penalises the estimated function if it contains high second differentials, would therefore be useful. Of course, if too much attention is paid to minimising the penalty terms (ν being too high) then an over-smoothed function arises, such as that in Plot 2 of Figure 2.1. It may well seem that not much is to be gained by using a regularisation term over simply estimating the c parameter for instance, given that in both situations the overall complexity has to be estimated (ν in one, c in the other). One advantage of penalty terms lies in the theory that already exists on them: certain types of problem are known to use certain types of regularisation term. Another advantage is their greater flexibility and, for us, the ease with which the theory can be extended to allow more specific forms of prior knowledge to be incorporated into the learning process.

Regularisation terms originally arose from work on computational vision (Poggio, Torre, & Koch, 1985), but there are now examples from cognition. These include Koh and Meyer (1991) and Busemeyer, McDaniel and Byun (1997), both of whom were modelling acquisition of continuous input / output mappings and why participants seek simple solutions first. Koh and Meyer argued that that the choice of function participants used to determine a mapping was based on a hypothesis testing procedure, coupled with Equation 19 as a penalty term to prevent over-fitting. Busemeyer *et al.* investigated how people acquire intervening concepts in a multivariate context. They found that

participants start off by trying to map individual input dimensions onto individual output dimensions, but then add an intervening concept if the environment suggested it. This was modelled by using a hidden layer network together with a regularisation term penalising solutions that use a large number of hidden units. The effect of the penalty term is that, as learning proceeds, the network settles on simple solutions.

Abu-Mostafa (1993, 1995) has developed a range of techniques for incorporating prior knowledge into neural networks based on altering the error functions. Abu-Mostafa describes prior knowledge as auxiliary information about the target function which can be used to guide the learning process, or "hints" as he calls them. A good example would be the invariances described earlier or knowledge that the target function is monotonically decreasing. The hints improve generalisation behaviour by placing a constraint on the allowable solutions, thus reducing variance. As mentioned previously, the hint does not increase bias because it is a valid property of the target function. Note that this means that training error may well increase because over-fitting (from those functions which disagree with the target function) has been reduced.

To incorporate the hints, two steps are needed. First, virtual examples need to be formed, which allow the algorithm to understand the information, and, secondly, extra error terms need to be introduced so that the virtual examples affect the weight solution. Virtual examples are pairs of examples which illustrate the hint, but say nothing about the real target function. For instance, for the monotonicity hint, a pair of input values (x, x') is chosen such that $x \leq x'$ and the input, x,

presented to the algorithm. The target value is simply the output from the estimated function of x', that is $f(x')$. The error function for the hint is:

$$e = \begin{cases} |f(x) - f(x')|^2 & \text{if } f(x) > f(x') \\ 0 & \text{if } f(x) \leq f(x') \end{cases} \qquad (20)$$

Thus, when monotonicity is realised, that is when f(x) is less than f(x'), error is zero, otherwise error is the difference between f(x) and f(x'). The error terms for the true examples and the virtual examples are then combined and gradient descent (or any other optimisation procedure) can then be performed in the usual way. In the actual training procedure, virtual and true examples are alternated so that the hint continues to be expressed as learning progresses; it is not the case that the hint is taught to the network before true learning takes place.

This method of incorporating knowledge has advantages and disadvantages. On the plus side, the implementational details do not have to be known in advance – the model is able to discover the representation for itself with a suitably defined error function. This in contrast to building knowledge into the structure (see Section 2.3.2), which requires that which weights to fix etc. are known in addition to the abstract hint itself. The main disadvantage is that extra computational demands are placed on the learning system by having to learn the 'virtual' examples, as well as real examples. There is clearly a trade-off the two issues here, so that when computational recourses are abundant, hints are at their most useful.

There are no psychological examples of the use of hints, as Abu-Mostafa describes them. Perhaps the closest idea is Heit's (1994, 1995) Integration Model, which assumes prior knowledge to be examples from similar categories added in to the to-be-learnt category (the two are not quite the same however, because exactly how the examples are to be incorporated is not specified in the Integration Model). Hints may well be useful to for psychological modelling however, because they take time to manifest themselves and there is an interaction between the knowledge and the environment (see Heit & Bott, 2000 for psychological evidence of this). Other methods, such as building in structure, take effect immediately and seem independent from learning. As the preceding paragraph stated, some situations will require at interactive knowledge effect, others need a static, immediate boost from their prior knowledge.

2.4 Summary and Conclusion

This chapter has examined statistical approaches to prior knowledge in relation to psychological findings. By demonstrating that knowledge can, and indeed *must*, be incorporated into models, it is hoped that that this review will encourage more psychological modellers to incorporate knowledge effects.

A more specific aim was to clarify what 'prior knowledge' is. This has been achieved by dissecting the literature into groups, centred around where in the learning process the effects occur. Thus, knowledge was first described as either reducing the complexity of a problem, or increasing the information available to the algorithm. Then, knowledge could be used either in the input, the structure, model order selection, the error functions, or any combination of the above. These distinctions work well for the statistical literature, and therefore at a computational level of analysis the breakdown is useful for psychology. However, it is much more difficult to classify the empirical work on prior knowledge in this way. On the complexity / information distinction, knowledge used for its information value involves asymptotic differences in training error, which would lead to differences in generalisation. Experimentally, this would imply that if the participant's asymptotic training error has been reduced after the introduction of the prior knowledge, then extra information is being used to eliminate hypotheses. Unfortunately, participants will always try to conform to what they perceive the experimenter wants, which generally means reproducing the training data regardless of what their prior knowledge indicates. This begs the question of whether non-asymptotic differences should be considered

indicative of informational knowledge, but if so, the distinction between the two concepts becomes even more blurred.

There are clearly problems with current experimental practice and the information / complexity distinction. This may be the fault of the experiments however, and not the dichotomy. For instance, much of the above discussion on prior knowledge has involved how to make the decision about when patterns in the environment reflect underlying structure and when they are random error. Most psychology experiments use small data sets and do not involve 'noise'. This greatly reduces the relevance of the informational knowledge and is consequently not an issue to the participant. Adding 'noise' and using environmentally appropriate data sets forces participants to compress the data set and decide which information to discard as error. It is these kinds of situations which might improve the psychological dissociation between information and complexity.

A similar point is whether the different approaches to incorporating knowledge will ever be psychologically discriminable. For example, even though the use of Abu-Mostafa's (1995) 'hints' are structurally different to using modularity to incorporate knowledge, is it possible to say that an organism is using one method, and not the other? Behaviourally, some methods do not look to be distinguishable. For instance, a regularisation term which allows a more complex solution as learning proceeds produces very similar behaviour to a constructive algorithm. On the other hand, even these techniques need discussing because differences at the *implemenational* level might arise through

neurophysiological findings – one might be more biologically plausibly than the other.

There are, of course, some aspects of prior knowledge which haven't been covered in this review. Most obviously, Murphy and Medin's (1985) ideas about the role of 'theories' or explanation-based reasoning in categorisation were not discussed. Murphy and Medin argued that similarity-based categorisation theories treat concepts as mere collections of features without providing an explanation for why features of categories hang together (their *conceptual coherence*). Instead, they suggest that features are interconnected within a rich relational structure, partly based on causal factors, and it is these 'theories' which guide the categorisation processes. For example, they argue that the features, "has wings" and "flies" are not just represented as independent attributes on a multi-dimensional space, but as functionally related properties. The idea that causality plays an important part in categorisation has also been empirically confirmed by, for example, Waldmann and Holyoak (1992), who showed that in a standard blocking paradigm, the redundant cue was only blocked if the cues correspond to 'causes', and not when they were perceived to be 'effects'. Similarly, Waldmann, Holyoak and Fratianne (1995) showed that by suggesting different causal structures in a learning task, the order in which people learnt linearly separable or non-linearly separable categories was reversed. Although these particular effects may be incorporated into similarity-based models by, for example, assuming a top-down adjustment of attention weights or the covariance matrix (but see Rehder, 1999, for a contrary view), a representation of *causation* seems beyond them: dimensions on multi-dimensional space are independent, by

definition. Further work could focus on how to combine the theory-based views with the similarity models, perhaps via the modular networks that have proved useful in reconciling 'rules' and 'exemplar' systems (see Section 2.3.2, *Modular architectures*).

Finally, this review demonstrates how intimately linked issues of task, representation, information, and learning speed are. The representation or structure the organism assumes will be influenced by what the task is, how many examples are available in the environment and how much time is available. Conceptually, a 'problem' doesn't exist in isolation from the other aspects of the learning environment. This implies that if the job of psychologists is to figure out how an organism performs a task, then the environment should play as much a part as the task itself.

Chapter 3[1]

In the previous chapter, different types of background knowledge were described together with methods of incorporating this knowledge into models. Using the bias variance distinction, it was argued that prior knowledge was necessary to reduce the number of examples needed for adequate generalisation and to prevent the possible combinations of examples from becoming prohibitively large. However, this solution raises another problem which is equally serious, namely that of how to select the appropriate knowledge from the large pool of potentially useful information. For example, consider trying to learn to identify a new artist's work. There are many sources of knowledge which might be useful, such as the school the artist adheres to, the materials they use, the influence of other artists on their work, and the ways that other artists can be identified. Given this vast array of knowledge, how can the most appropriate knowledge be selected? Is it really a help to shift the problem away from breaking down a large space of possible categories to selecting relevant information from a large space of possible prior knowledge?

On the face it, the knowledge selection problem appears quite daunting, perhaps even insoluble. It is worth noting, however, that human beings offer an "existence proof": there is no question that background knowledge is used in the formation of concepts (see Heit, 1997, and the previous chapter) and therefore must be selected in some way. Furthermore, Bayesian statistical methods have

[1] The modelling work reported Section 3.1.2 of this chapter has been published in Heit and Bott (2000).

always utilised multiple prior distributions with the assumption that new observations alter (or select) the degree of belief in these prior hypotheses (see also Heit, 1998). In short, there is no reason to consider that knowledge selection cannot be investigated empirically, or that modelling is unsuitable.

Most previous experiments on background knowledge have tended to avoid the problem of how the relevant knowledge is selected by making it obvious which knowledge is relevant, or at least not concerning themselves with how the selection process took place. For example, in Dienes, Altman and Gao's (in press) transfer experiments on artificial grammar, it was implicitly understood that any knowledge acquired in the first phase of the experiment should be applied in the second phase. In Heit's (1994) studies on integration effects, participants were asked more explicitly to make judgements based on specific prior knowledge and some observed examples. In contrast to these experiments, both Murphy and Allopenna (1994) and Heit and Bott (2000) have left the decision about which knowledge to select far more ambiguous. Because the simulations and experiments presented in this chapter extend the work carried out on these latter studies, they will be described in some detail.

Murphy and Allopenna (1994) asked participants to learn to classify examples as either "Category 1" or "Category 2", thereby denying them explicit guidance as to the relevant knowledge needed. However, the contents of the observations themselves proved useful in selecting the appropriate knowledge. The examples consisted of descriptions of vehicles (or buildings), such as "Made in Africa, lightly insulated, drives in jungles". On reaching a learning criterion, it was

established that participants had mapped the empirical data onto their background knowledge. For example, participants had used information such as "drives in jungles", to rule out the possibility that knowledge about snowmobiles (another one of Murphy and Allopenna's categories) was relevant.

Heit and Bott (2000) extended these experiments by examining performance changes during the learning process. The idea behind this was to show that the effects of knowledge could become more pronounced as learning took place, contrary to what most theories would predict. For instance, Heit (1995) showed that as more examples were observed, the effect of prior knowledge diminished - participants made judgements based more on the observed data. We also wanted to collect more data to enable us to model the experiments. In Experiment 1 of our study, participants were told that they were going to learn about two types of buildings, Doe and Lee buildings. The Doe and Lee categories corresponded loosely to a "church" versus "office block" distinction, although participants were not informed of this. They were then presented with a series of descriptions of buildings, together with the appropriate category label. The descriptions were made up of several different types of features: Critical features; Filler features; and Individuating features. The Critical features were designed so one item from each was typical of a church, whereas the other was typical of an office block. For example, the 'lighting' feature consisted of the values "Lit by candles" and "Lit by strip lights", indicating a church and an office block respectively. On the other hand, Filler features were designed so that either value from each pair could fit into a church or an office block equally well. For instance, one Filler feature pair was "Designed by a local architect" versus "Designed by an

international architect". A full list of Filler and Critical features is shown in Table 3.1. We expected that as participants viewed more and more exemplars,

Critical Features	**Filler Features**
has steeply angled roof	near a bus station
has wooden furniture	designed by a local architect
has an interesting structure	has gas central heating
old building	has steel piping
quiet building	has a foyer
lit by candles	near a river
ornately decorated	has a lightning conductor
built with stone	has grey phones
has a flat roof	not near a bus station
has metal furniture	designed by an international architect
has a repetitive structure	has electric central heating
new building	has copper piping
busy building	doesn't have a foyer
lit by fluorescent light	not near a river
blandly decorated	doesn't have a lightning conductor
built with metal and concrete	has blue phones

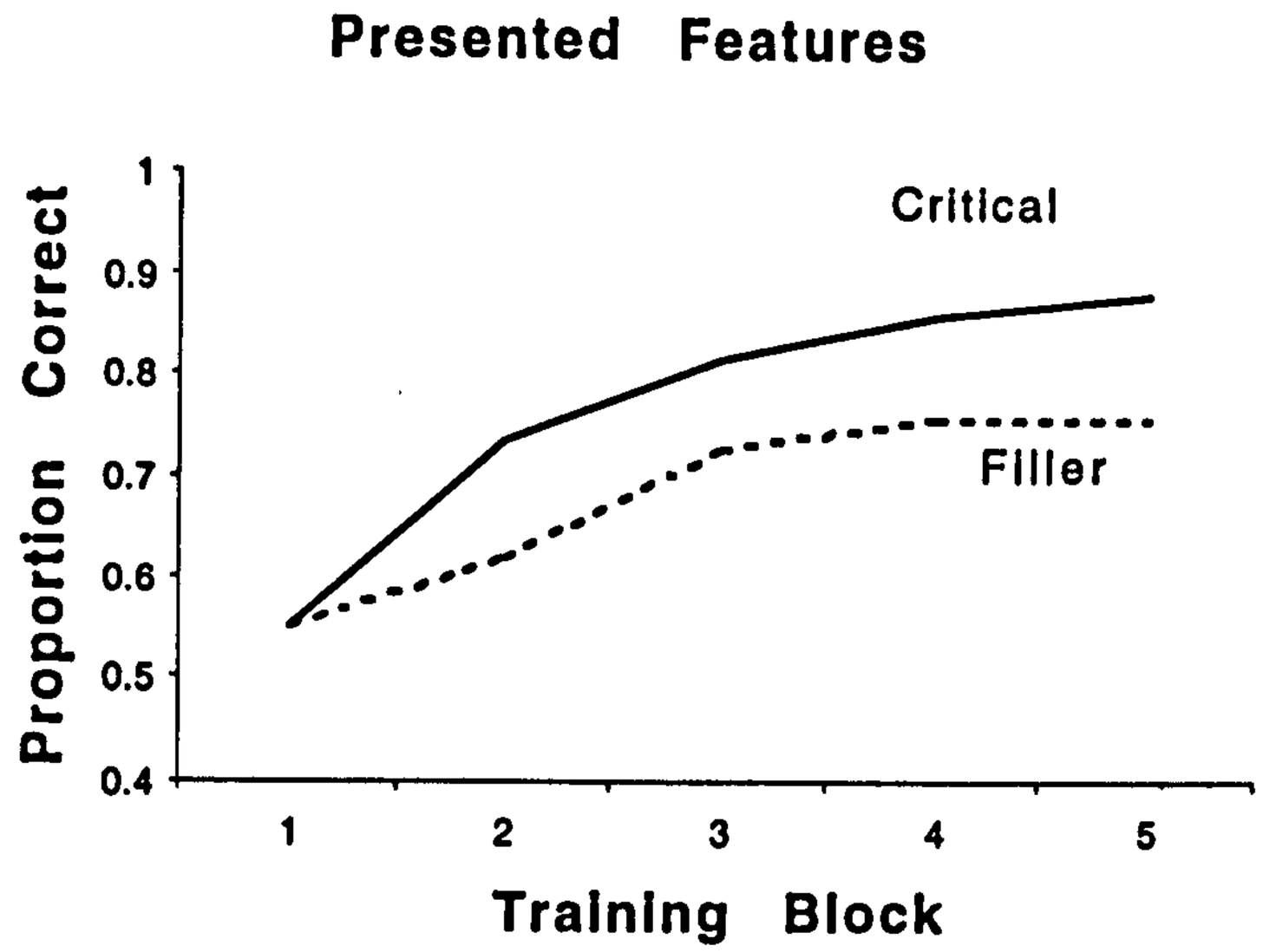
Table 3.1 Critical and filler features for building stimuli.

their prior knowledge would improve their performance on Critical features (because they would gradually realise that Doe buildings corresponded to

Churches, say) but not on Filler features. Individuating features were designed to slow participants down but played no other part in the design. A sample description would be {Lee building type, Photographer: T. Evans, designed by a local architect, has wooden furniture, Builder: N. Stewart, has a steeply angled roof, Surveyor: A Ferraro, near a bus station}, which is of the form {Label, individuater, filler, critical, individuater, critical, individuater, filler}.

From the 8 pairs of critical features, 4 pairs were randomly assigned to presentation frequency one. Each feature in these pairs were presented in one description per block, either Doe or Lee. Two pairs were assigned to presentation frequency 2, and these were presented in 2 descriptions per block. Finally, 2 pairs of features were not presented at all in the study blocks (but they were in test blocks). There were 5 training blocks, each followed by a testing block consisting of questions asking whether an individual feature was likely to belong to a Doe or a Lee building. All features were tested, including the 40 individuating features and those critical and filler features which were not presented during training.

The results confirmed the hypothesis that knowledge had an increasing effect as more blocks were experienced. One way of seeing this is by examining those features presented during training, collapsed across frequency, as displayed in the upper panel of Figure 3.1. At the start of learning, critical and filler features are known equally well. However, as learning continues, the gap between the two curves increases, as confirmed by a significant interaction. The knowledge



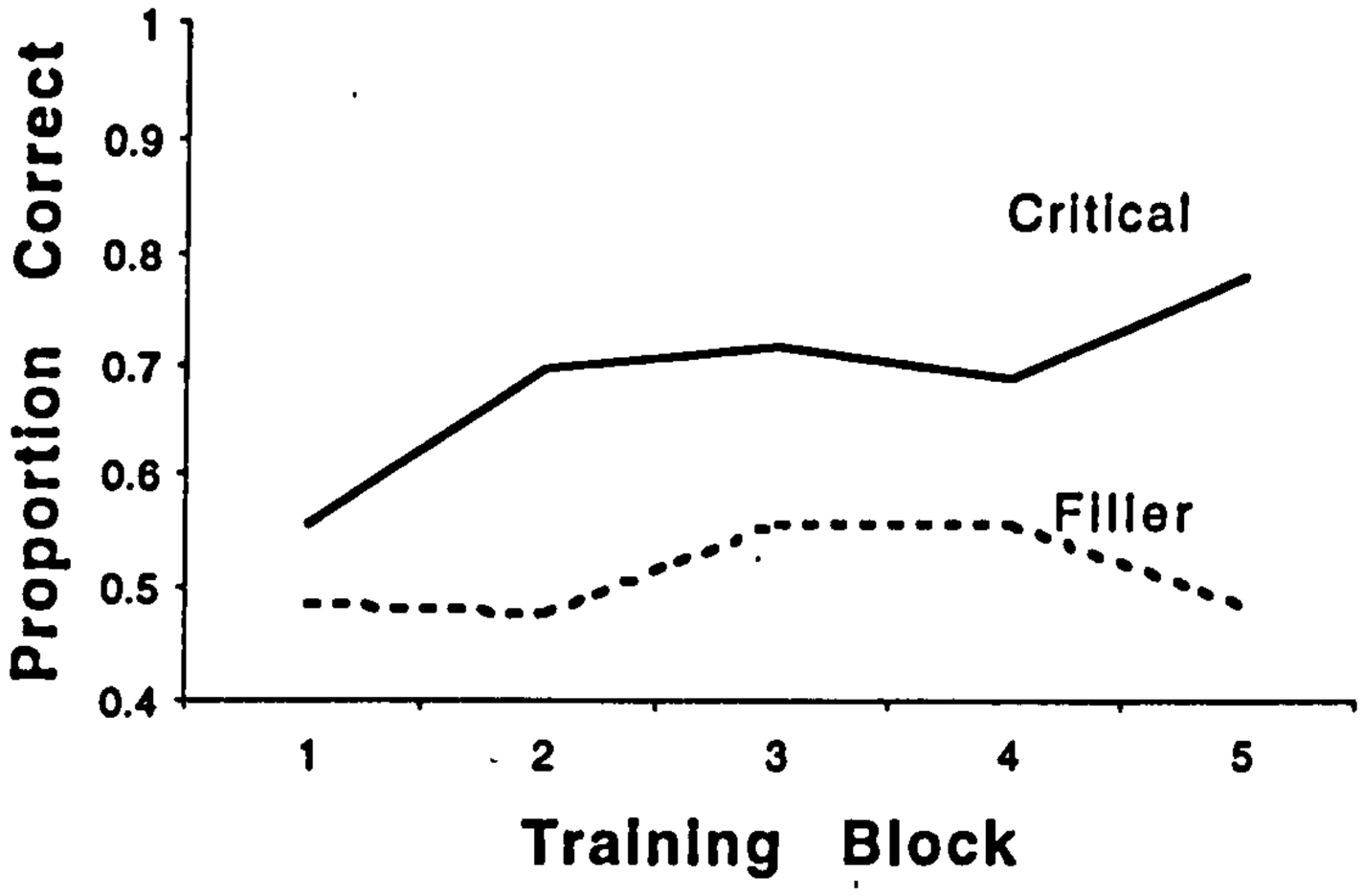


Figure 3.1 Results from Heit and Bott (2000).

that Doe's might be churches and Lee's office blocks only benefits critical features, and this effect is only observed in later learning. A similar pattern can be seen in the lower plot of Figure 3.1, in the data from unpresented items. It is worth clarifying that filler features which haven't been presented in the study phase cannot be classified above chance; knowledge of the mapping to church and office blocks does not help to classify the feature "designed by a local architect" for instance. On the other hand, even if "is lit by candles" has not been presented, knowledge that Doe is a church can easily lead to appropriate classification. The plot of unpresented items confirms this, with percentage correct for critical items gradually increasing as the experiment continues.

One slightly surprising result was that the frequency manipulation seemed to have no effect. It is tempting to relate this to Murphy and Allopenna's (1994) study, which demonstrated reduced sensitivity to prior knowledge features. However, Heit (1994, 1995, 1998) found robust frequency effects in prior knowledge so it would be wrong to say that people are not sensitive to frequency in categorisation involving prior knowledge. Further, it is clear that a manipulation of, say, 20 presentations to 2, would have an effect on percentage correct. Although an interesting result, further work is required before firm conclusions can be drawn on this issue.

In summary, the effects of prior knowledge were found to increase through learning, as manifested in the interaction between critical and filler items. As mentioned above, Heit (1995) found the contrary, that is, a reduced effect of knowledge as learning progressed. The important difference between the two

studies was that Heit and Bott (2000) used category labels which didn't guide the choice of prior knowledge, whereas Heit (1995) asked questions which were far more explicit about which knowledge was relevant. In Heit and Bott therefore, it can be concluded that examples were needed to select the appropriate knowledge at the beginning of learning. The new work presented in this chapter simulates the Heit and Bott experiment in an attempt to provide an underlying theory. These simulations lead to new empirical predictions which are also tested and described here.

3.1 The Baywatch Model

The approach to knowledge selection presented here has some parallels to the mixture-of-experts architecture (Jacobs, Jordan, Knowlan & Hinton, 1991, as described in Chapter 2), but instead of using modules with different structures, modules with different pools of pre-trained knowledge were used. Therefore this method also has some relations to techniques that insert prior knowledge directly into networks. The model, illustrated in Figure 3.2, can be described as having one module or set of weights for strictly empirical learning. These weights do not get any pre-training. Then the model also has a set of experts which are pre-trained to recognise different known categories. For example, a network for learning about buildings might have experts which can recognise different kinds of buildings such as churches, office blocks, restaurants, and schools (only two of these expert modules are illustrated in Figure 3.2). The model will be referred to as the Baywatch model because it combines a general Bayesian approach to selecting among multiple sources of prior knowledge with an empirical learning component.

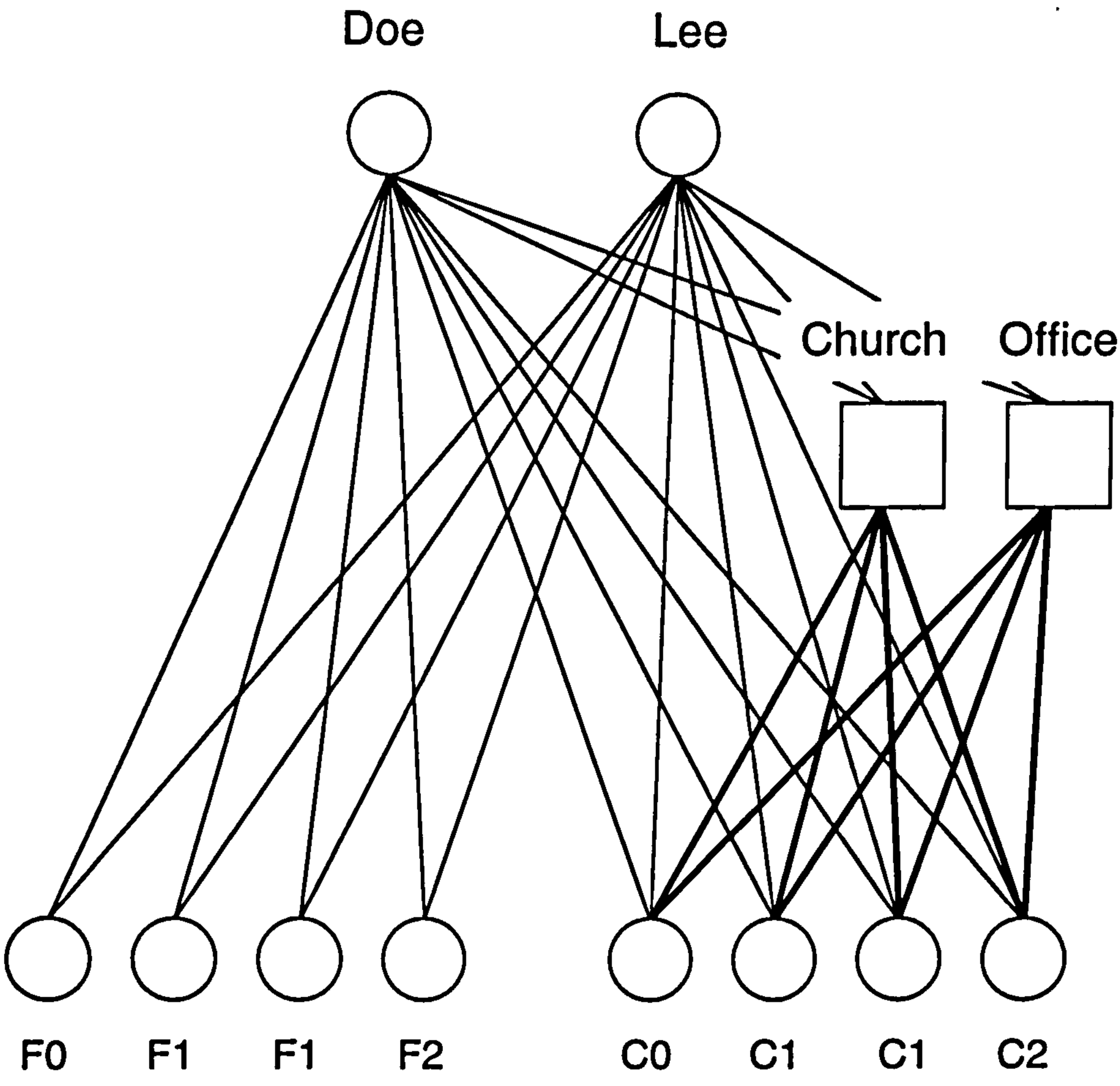


Figure 3.2 Illustration of Baywatch model. Fixed weights are shown by connections in bold.

The Baywatch model is a feedforward network where the input units represent the individual features and the output units represent the Doe and Lee category nodes. The two hidden units correspond to two expert modules, or prior knowledge category nodes (PK nodes). The four input units on the left of Figure 3.2 represent filler features, and the four inputs on the right represent the critical features. The only difference between the two types of features is that the filler features are only connected to the output nodes, whereas the critical features are connected both directly to the output nodes and indirectly to the output nodes via the prior knowledge nodes. The difference between filler and critical features in the model reflects our assumptions about how learning would take place in our experiments. Consequently, we required filler features to be learned directly without the help of prior knowledge, whereas critical features were to be learned both directly and by a mediated connection through prior knowledge. The connections between the critical features and the PK nodes have fixed weights, so that values of critical features of the stimuli that correspond to church features would activate the church PK node, and likewise critical features of the stimuli that correspond to offices would activate the office PK node. It is assumed that these fixed weights would correspond to prior knowledge about familiar characteristics of churches and office blocks, learned through ordinary means of association. The PK nodes have threshold functions, so that if any church feature, say, steeply angled roof, is presented, then the church PK node will be activated. The activation from the PK node would then be propagated to the output units.

In contrast to the connection weights between the critical features and the PK nodes, the other weights in the network are learnable through gradient descent on the error between the desired output of the network and the actual output. Adjusting the weights from filler units and the critical units to the output units allows the features to be associated with the category nodes in the empirical learning module. Note that if these were the only weights in the network, there would be no difference between the two types of features. Finally, there are adjustable weights between the PK nodes and the category nodes. These represent the participant's capacity to associate known categories, say churches and office blocks, with the new categories, Doe and Lee buildings. This part of the network can be seen as addressing (at least in part) the knowledge selection problem, because here the network is learning to select from already known categories and apply this knowledge to judgements about new categories.

3.1.1 Technical details

The input units can take on the values {+1, 0, -1}, which correspond to the Doe value of a feature, the feature not being present, and the Lee value of a feature respectively. For instance, if the feature is the lighting feature (see Table 3.2), then a -1 value would mean "lit by candles" value, a 0 would correspond to not presenting the feature at all, and a +1 would mean "lit by fluorescent lights." The two output units vary continuously between -1 and +1. One output unit corresponds to the Doe category and the other to the Lee category. The activation on each category was given by the weighted sum of its inputs. This activation was then converted into a probability measure using the logistic

transformation given in Gluck and Bower (1988, Equation 7). If a Doe exemplar is presented during training, the teaching values for the Category nodes are +1 on the Doe node and -1 on the Lee node (see Table 3.2). These values would be reversed for a Lee training example.

Filler Features				Critical Features				Desired Output	
1	1	0	0	1	1(-1)	0	0	1	-1
1	0	1	0	1	0	1	0	1	-1
-1	-1	0	0	-1	-1(1)	0	0	-1	1
-1	0	-1	0	-1	0	-1	0	-1	1

Table 3.2 Structure of the Training Data. Figures in parentheses correspond to the Incongruent feature values (see Section 3.1.3).

Critical features are connected by fixed weights to the PK nodes. As can be seen from Figure 3.2, these were connected so that if the Lee value (-1) of a feature is presented, this lead to positive activation on the church PK node (because Lee buildings would correspond to churches), and a negative activation on the office node. The output of a PK node was a threshold transformation of the weighted sum of its inputs, such that the output was 1 if the sum was greater than or equal to 1, and 0 otherwise. All of the weights in the network were adjusted according to the standard delta rule (e.g., Gluck & Bower, 1988).

As the introduction suggested, the model can also be construed in the modular framework put forward by Jacobs *et al.* (1991). To see this, consider a slight variation of the model shown in Figure 3.3.

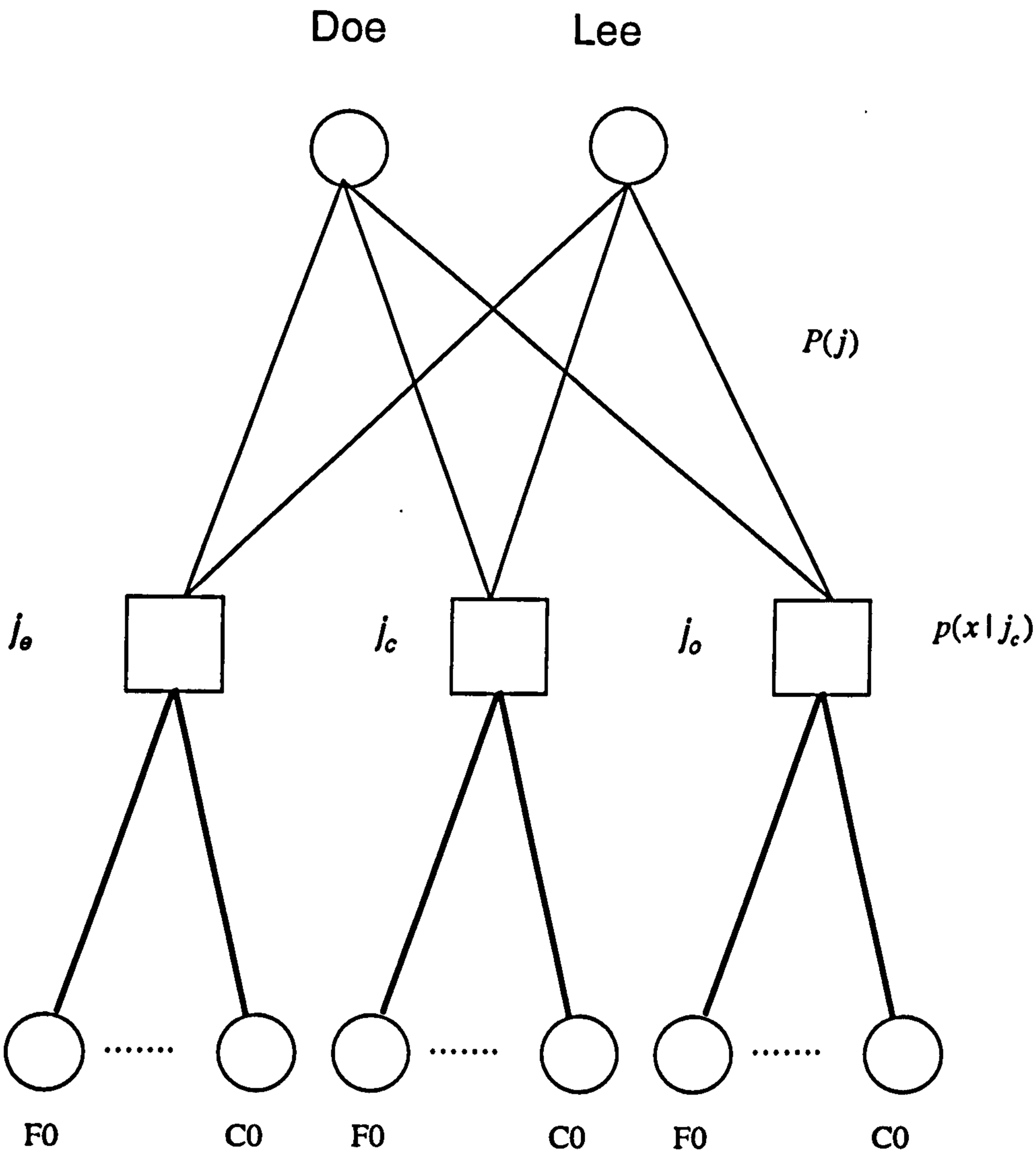


Figure 3.3 Alternative version of Baywatch. Fixed weights are shown by connections in bold.

Instead of PK nodes, the hidden units are now labelled as 'experts' to be more like the notation used in the modular approach: j_c for the Church expert; j_o, for the Office Block expert; and a new Empirical expert labelled as j_e. The previous version of the model did not have the Empirical node, but incorporating it aids

the exposition and only minor differences in simulation results arise. This hidden node is connected to the input dimensions in such a way that the presentation of any input feature will activate it. As with the other weights on the first layer of the network, these are fixed weights. Another notational difference is that each hypothesis now has connections to all the features, which means that the model can be thought of as being set in an 8-dimensional space (four filler dimensions and four critical dimensions). As before however, the Church and Office hypotheses are only activated when vectors fall in a certain area of the space, whereas the Empirical hypothesis gets activated regardless of the vector. Furthermore, all these input weights are set prior to learning and cannot be altered.

The mixture of experts framework assumes that the output of each expert corresponds to the probability of generating the test item from that expert, that is, $p(\mathbf{x} \mid j)$. For the Church and Office Block experts for example, these are 1 if the test value falls in the relevant portion of the space, 0 otherwise. The weights leading from the expert to the output nodes are the *mixing* coefficients (or priors for each expert), $P(j)$. This means that after the Doe / Lee category node has summed the expert-conditional densities, class assignment can be made on the basis of the probability density function:

$$p(\mathbf{x} \mid C_k) = \sum_{j}^{M} p(\mathbf{x} \mid j) P(j) \qquad (1)$$

where M is the total number of experts (three, in this case). In other words, the final density function is a linear combination of the outputs of the experts, or a

mixture distribution. A distribution is calculated for each output category, Doe or Lee, and these can then normalised to produce the equivalent of proportion correct.

This alternative description of the model allows it to be seen in the context of more complicated algorithms for classification, such as those described in Chapter 2. However, to be consistent with most psychological theories, the more traditional, connectionist approach introduced earlier will be used for the rest of the chapter.

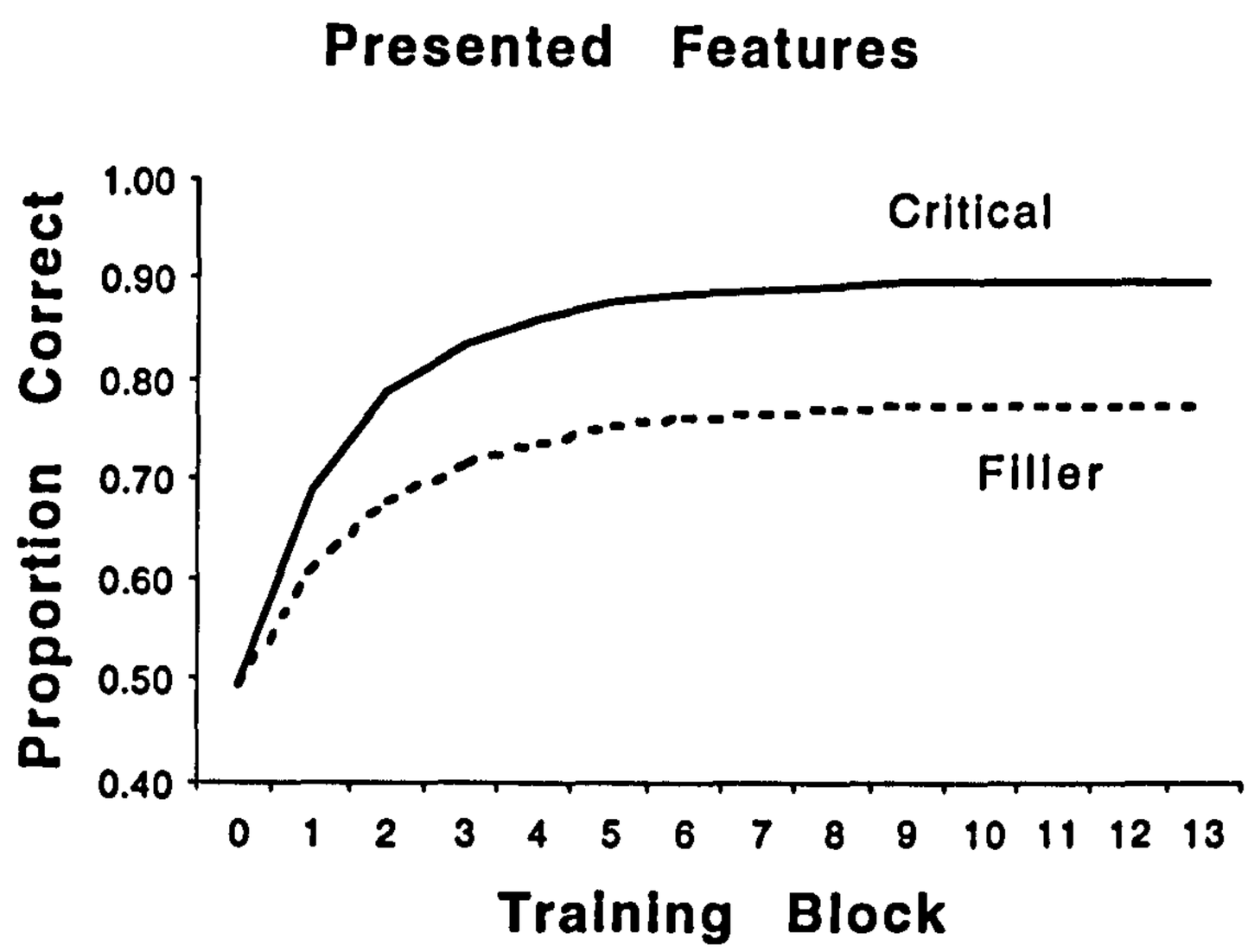
3.1.2 Simulation of Heit and Bott (2000) Experiments

The network shown in Figure 3.2 was used for these simulations. Training was for a total of 13 epochs, with the learning rate in the delta rule set at 0.1 and the probability mapping constant for the logistic transformation function set at 7.0 (both values were derived from an informal sampling of the parameter space). The training stimuli consisted of four examples of buildings, two Doe exemplars and two Lee exemplars, which are shown in Table 3.2. The first two rows are the Doe buildings and the second two the Lee buildings. Note that the fourth features in the critical feature section and in the filler feature section always have a value of zero. These features correspond to those that were never presented to the subjects in the experiments. Following each training epoch, the network was tested on the individual features by presenting a vector of all zeroes except for the particular feature of interest, which had a value of either +1 or -1. The results of the simulations are displayed in Figure 3.4, with the proportion correct on the test set shown as a function of the number of learning epochs and feature type.

The top panel shows the model's predictions for presented features. The responses to features presented once per epoch and twice per epoch are pooled together, as they were for the Heit and Bott (2000) experiments. The bottom panel shows predictions for features that had not been presented during training. The predictions fit well with the main results of the experiments. Critical features were learned more quickly than filler features, and Critical features that hadn't been presented were responded to more accurately than chance, whereas filler features which hadn't been presented were at chance level.

The first result can be explained in terms of the extra connections from critical feature inputs to the output units, mediated by connections through the PK nodes. As the network progressively learned which sources of prior knowledge correspond to the Doe and Lee categories, responses on critical features were derived both from the empirical learning module and from prior knowledge. In addition to these two paths of influence on the category outputs, the other advantage for critical features over filler features is that there are two paths of learning, in effect leading to twice as much updating of weights after a particular learning trial.

A similar advantage for presented critical features over presented filler features might be obtained without any PK nodes at all, by simply increasing the learning rate on the critical features relative to the filler features. However, that scheme



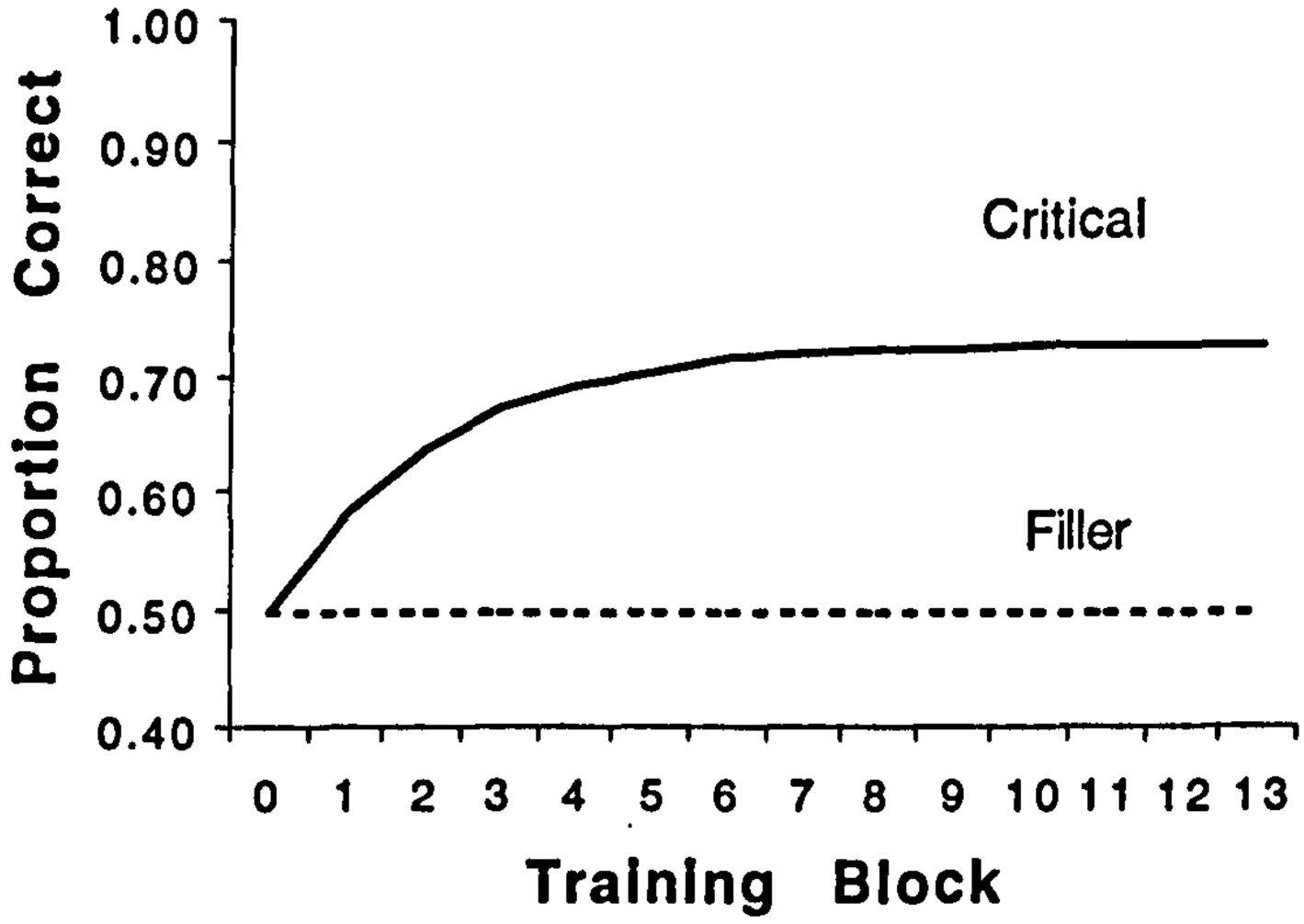
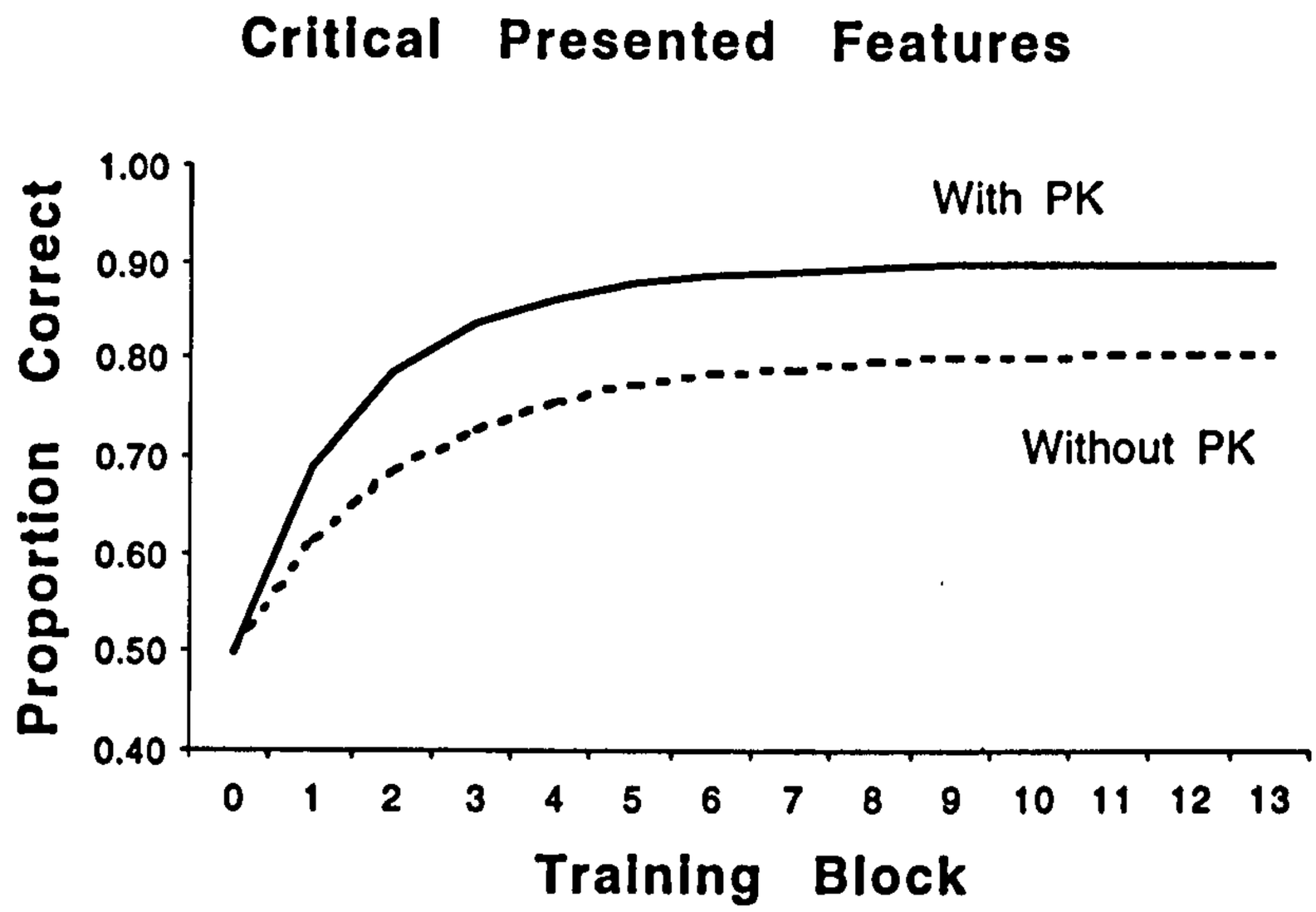


Figure 3.4 Simulation of Heit and Bott (2000).

would not predict any advantage for non-presented critical features over non-presented filler features. In the Baywatch model, for non-presented critical features and filler features, the weights leading from the input units directly to the output units remain at zero throughout learning. Because this is the only way the filler features can activate the output units, their accuracy stays at chance level. In contrast, the non-presented critical features have another route to the category units, through the PK nodes whose weights are adjusted when any critical feature are presented. Therefore the PK nodes are critical to the Baywatch model's predictions on non-presented critical features.

To provide a better idea of how the Baywatch model uses prior knowledge, the simulations were run without any PK nodes, for comparison. Figure 3.5 shows simulated predictions on presented items, comparing versions of the model with and without prior knowledge. For critical features, in the top panel, it can be seen that the prior knowledge does not have any influence initially on judgements; the model acts the same way with or without PK nodes. However, the beneficial effect of prior knowledge for critical features increases over the course of learning, as the network with PK nodes learns which categories to connect with its prior knowledge. In the bottom panel of Figure 3.5, there is evidence of a slight detrimental effect of prior knowledge on the learning of filler features. This result can be explained as a kind of overshadowing effect, in which knowledge of some highly predictive cues can reduce learning on other predictive cues. As a consequence of the delta rule, when the network learns to predict the outputs increasingly well from the critical feature inputs, learning on



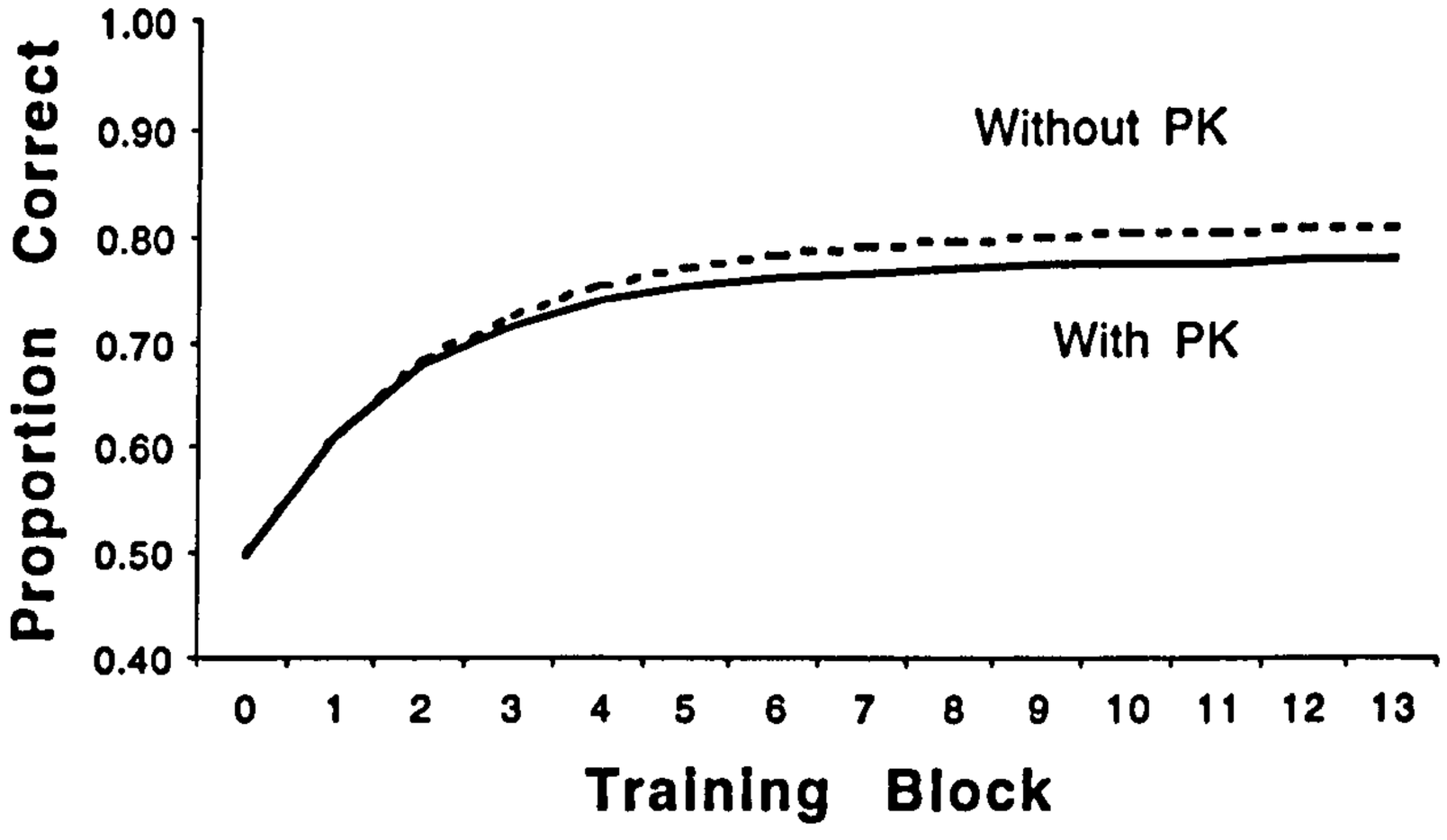


Figure 3.5 Predictions with and without Prior Knowledge nodes.

the filler features will increasingly be disadvantaged. This aspect of the model is discussed further when evaluating the model in general.

In summary, the Baywatch model captures many of the important features of the two Heit and Bott (2000) experiments on knowledge selection in category learning. At the start of learning, the model is not influenced by prior knowledge, because it does not know which past categories are useful for making predictions about the Doe and Lee categories. But as observations are made, the model is able to select relevant prior knowledge to be used for judgements about the novel categories. This influence of prior knowledge leads to a persistent advantage for critical features over filler features. Although there are several questions which could be raised about the model, such as the overshadowing discussed above, perhaps the most fundamental is how the model might scale up: there are only two PK nodes in the simulations, whereas people might be expected to bring far more potentially useful categories to the experiment. This next section describes a series of simulations which examine the issue and generally look at how the scope of the model might be expanded.

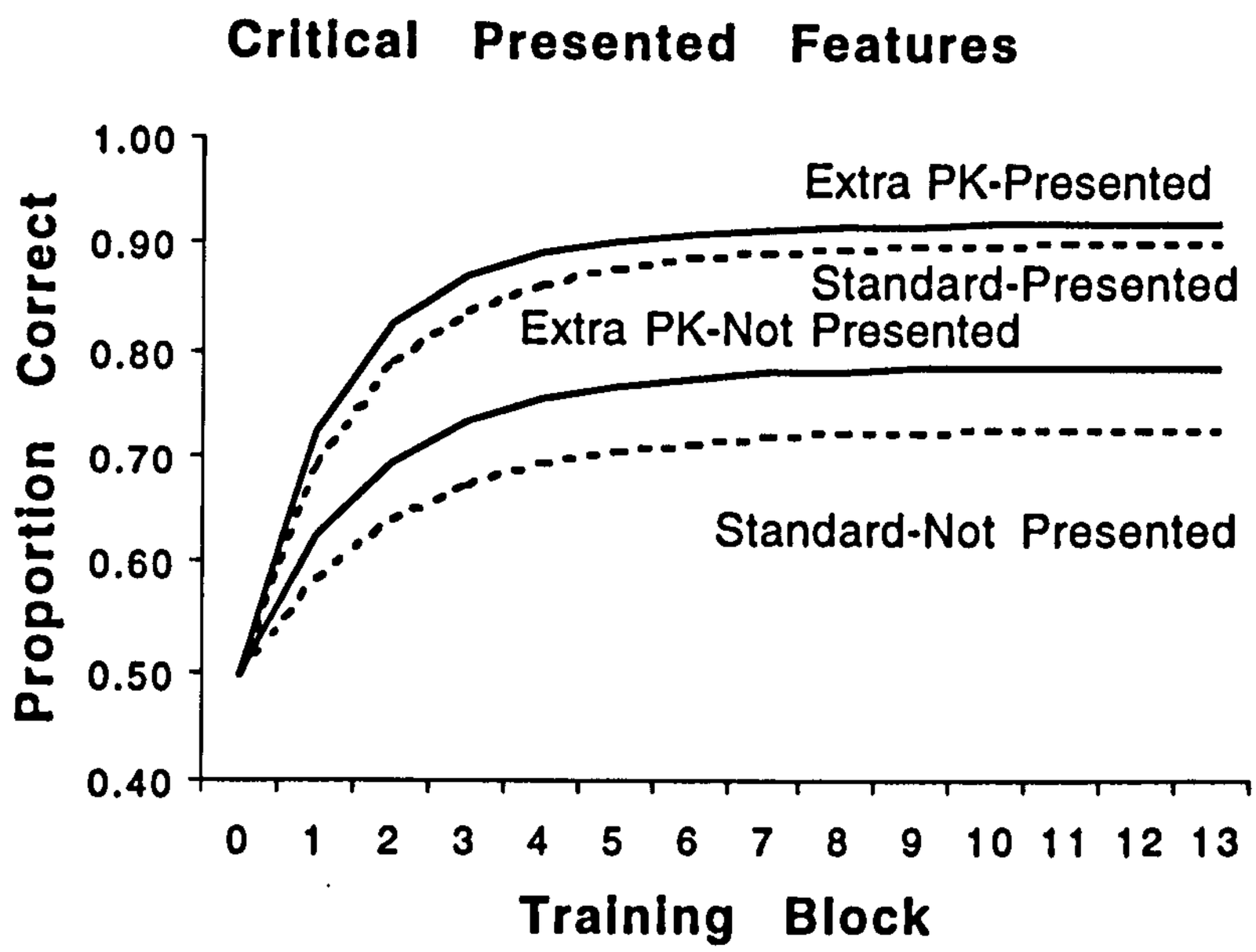
3.1.3 Further Simulations

Altering the number of PK nodes

Heit and Bott (2000) describe three different types of PK nodes which might be added to the network. First, completely irrelevant prior knowledge nodes might

be added, which have little or no connection to the input stimuli. For example, there could be prior knowledge nodes for space stations, igloos, tents, and cave dwellings, added to the network, but these nodes would be hardly activated by the inputs. For example, an input feature such as "lit by fluorescent light" would not be strongly associated with these categories, according to prior knowledge. Therefore, adding PK nodes that are irrelevant to the stimuli would not affect the results of the simulations very much.

Second, additional PK nodes that are similar to the existing PK nodes might be added. For example, a PK node corresponding to cathedrals would entail much of the same connections to inputs as the church node. Likewise there might be similar PK nodes for industrial parks and office buildings. In further simulations, we added a cathedral PK node that had two connections to the critical features for churches (to the critical feature presented twice and the non-presented critical feature) and an industrial park PK node that likewise was connected to two critical features for office buildings. The results are shown in Figure 3.6, comparing the original simulations with two PK nodes to the new simulations with four PK nodes. Inserting the two additional PK nodes improved performance on those critical features that now had two paths for knowledge-directed learning. However, inserting PK nodes did worsen performance on filler features, because the additional reliance on critical features led to some overshadowing of filler features. Likewise there was a slight decrement on performance (not shown in Figure 3.6) on critical features that differed within a pair of PK nodes (e.g., features that were true of office buildings but not industrial parks). Still, to the extent that sources of prior knowledge were



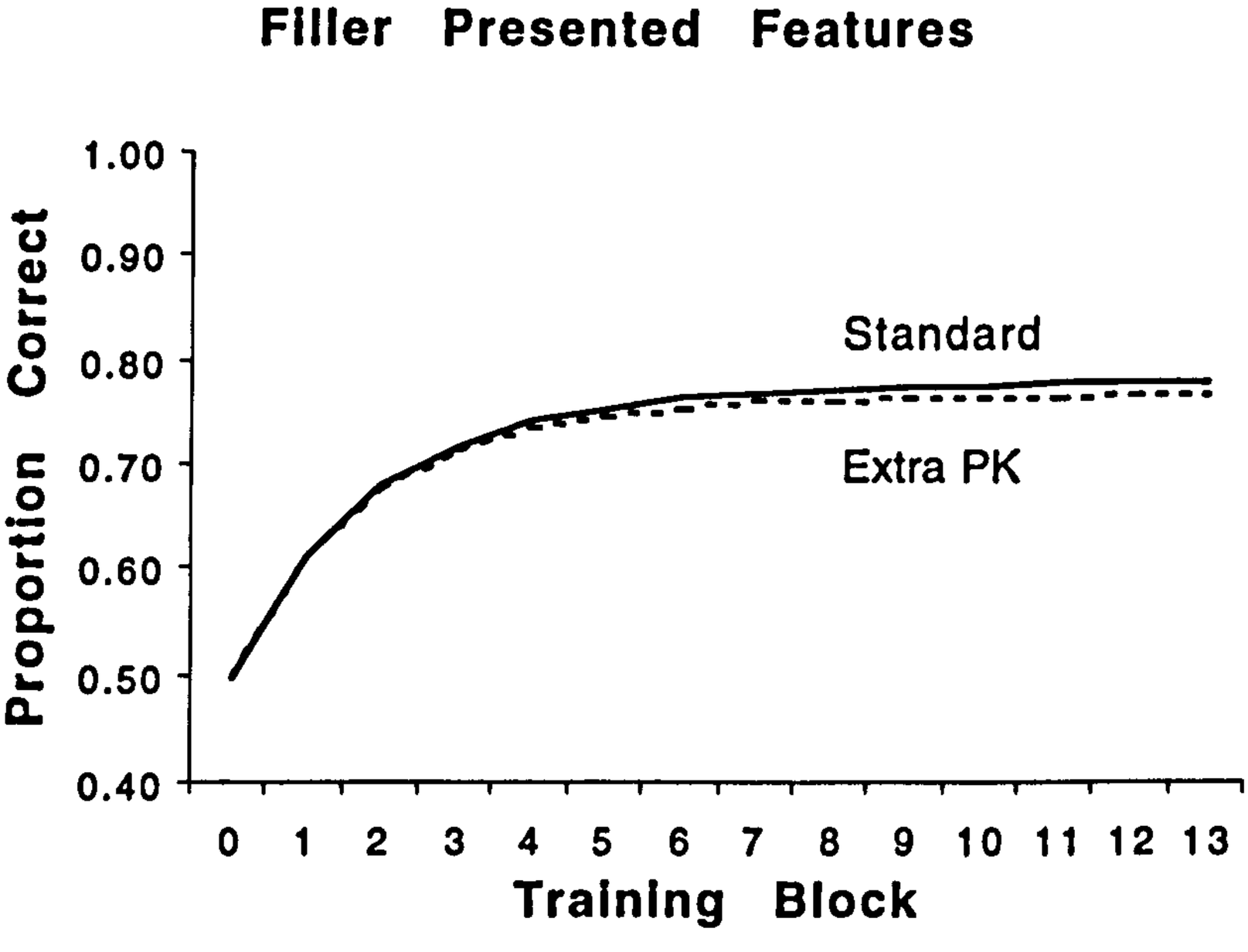


Figure 3.6 Results of the simulations involving extra PK nodes.

mutually supporting, having multiple sources of prior knowledge helped performance. Generally speaking, we did not find that adding additional, similar PK nodes led to a knowledge selection problem. This results highlight an interesting question about the Heit and Bott (2000) experiments. Although we observed better performance on critical features than filler features, due to increased use of prior knowledge, the results themselves do not indicate which prior knowledge was being retrieved. Some subjects could well have been retrieving knowledge about cathedrals rather than churches, or industrial parks rather than office buildings. Indeed, informal debriefings of subjects revealed some variety of responses to questions about what the experimental stimuli were like in the real world.

Third, "malicious" prior knowledge nodes could be added to the network, for example, prior knowledge about some kind of building that is half-church and half-office block. This was simulated by creating an extra PK node (a 'Choffice' node) and linking it up to the two single presentation features. The weights were set such that the Office value of one feature activated the node, and the Church value of the other activated the node. Although it was initially expected that malicious PK nodes would hurt performance, very few negative effects arose in practice. This was because Choffice is associated with both Doe and Lee on different items in the training set and consequently, very little weight built up on the PK to Output nodes. Again, no knowledge selection problems arose when malicious PK nodes were added.

Incongruent training exemplars

An alternative method of investigating "malicious" prior knowledge is to make one of the Critical features incongruent with the others, the result being that a Church feature would appear in the same item as an Office feature. For example, the features "lit by candles" and "new building" would appear together. Notice that without the effects of prior knowledge, these features would not be incongruent. The simulation was carried out by switching the sign of one Critical feature value in training (see the values in parentheses, Table 3.2), making it incongruent with the double-presentation feature and the other single-presentation Critical feature. There were several noteworthy effects of this manipulation, as displayed in Figure 3.7. The first is that the effects of knowledge have been reduced overall, as measured by a reduction in the difference between Critical Congruent and Fillers, and a drop in the accuracy of the Unpresented Critical features. Secondly, the Critical Incongruent features are learnt worse than even the Fillers. These effects are because the Critical Incongruent feature drives the weights on the PK to category in the opposite direction to the other items. This means that at the end of learning, the PK weights are less developed and consequently provide less of an advantage to the Critical Congruent features. Furthermore, Critical Incongruent suffer as a result of the PK nodes working in the wrong direction, although they still have the weights on the empirical side of the network to provide some form of learning.

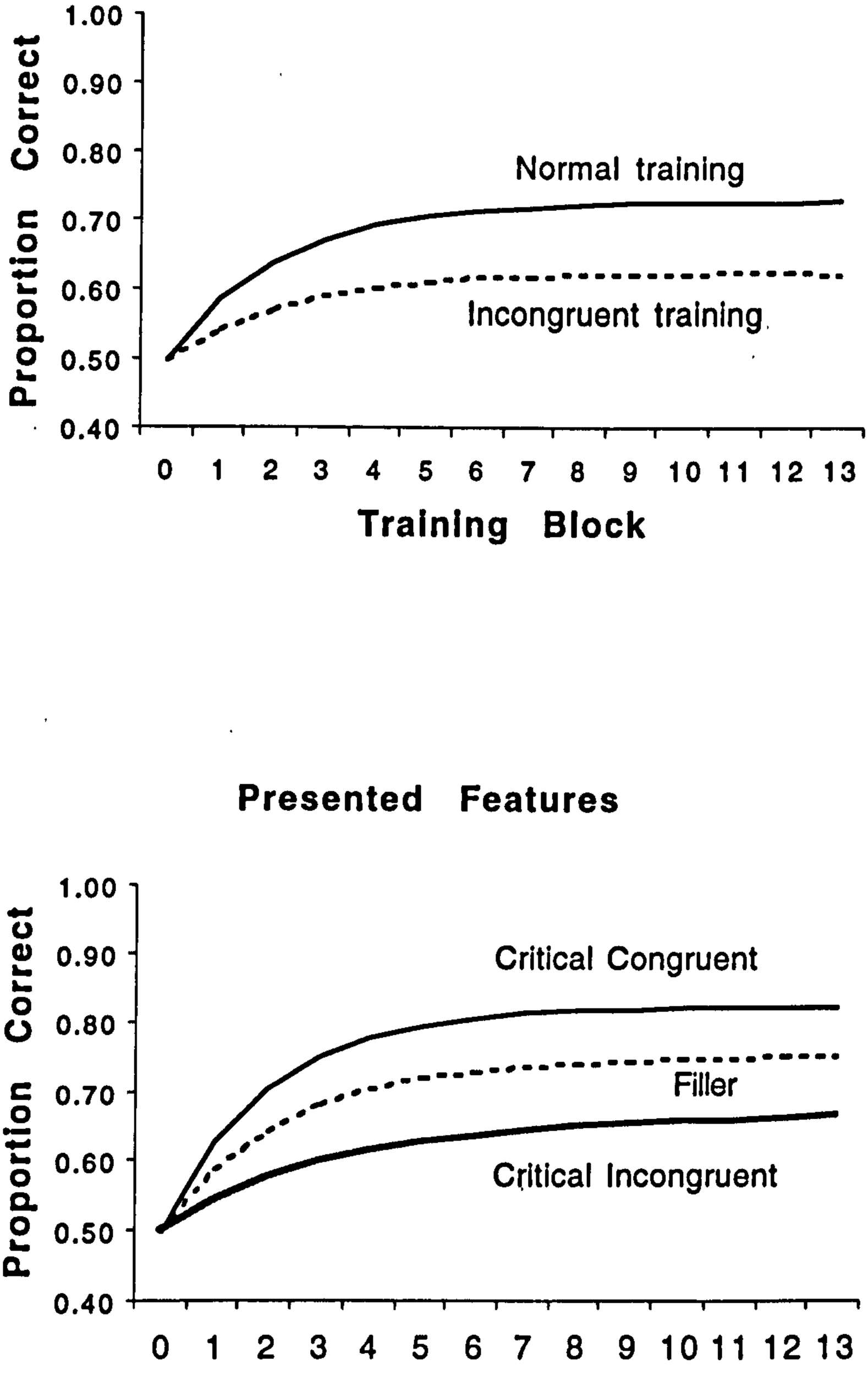


Figure 3.7 Predictions from training with incongruent features.

Providing the network with flexible, pre-programmed weights

Chapter 2 described work by Giles and Omlin (1993) on providing networks with pre-programmed weights to aid learning. Unlike the pre-programmed weights in Baywatch (from input to PK nodes), Giles and Omlin allowed their weights to be fully trainable. In the simulations presented in this section, this form of representing knowledge is experimented with. Specifically, the network was given weights representing knowledge on the relationship between Church / Office categories and the Doe and Lee output nodes.

Four simulations were carried out: two which involved 'true' hints, that is, knowledge which turns out to be correct; and two which give 'false' hints, or incorrect knowledge. An example of a true hint provided to a participant might be, "Lee buildings are like churches" when the stimuli suggests they are. A false hint for the same set of stimuli would be, "Lee buildings are like office blocks", when, in fact, they are like churches. For each validity type, there are 'strong' and 'weak' hints. Strong hints are incorporated into the network by setting the Office to Output node weights to high magnitudes (+/-1), whereas weak hints are set at lower magnitudes (+/-0.5). The sign of the weight indicates the veracity of the hint. All other PK to Output weights are set as zero. As an example of a strong, true hint, consider Figure 3.8. The Office to Lee weight is set at +1, the Office to Doe weight is set at -1, and the others are set to zero. In other words, Offices are Lee, and they are not Does, and nothing is said about Churches. The training stimuli would also indicate that Lees are Offices because this hint is

'true'. Note that because a simulation with a hint for Lee items produces analogous results to a simulation with a Doe hint, only the results from Lee hint simulations are reported.

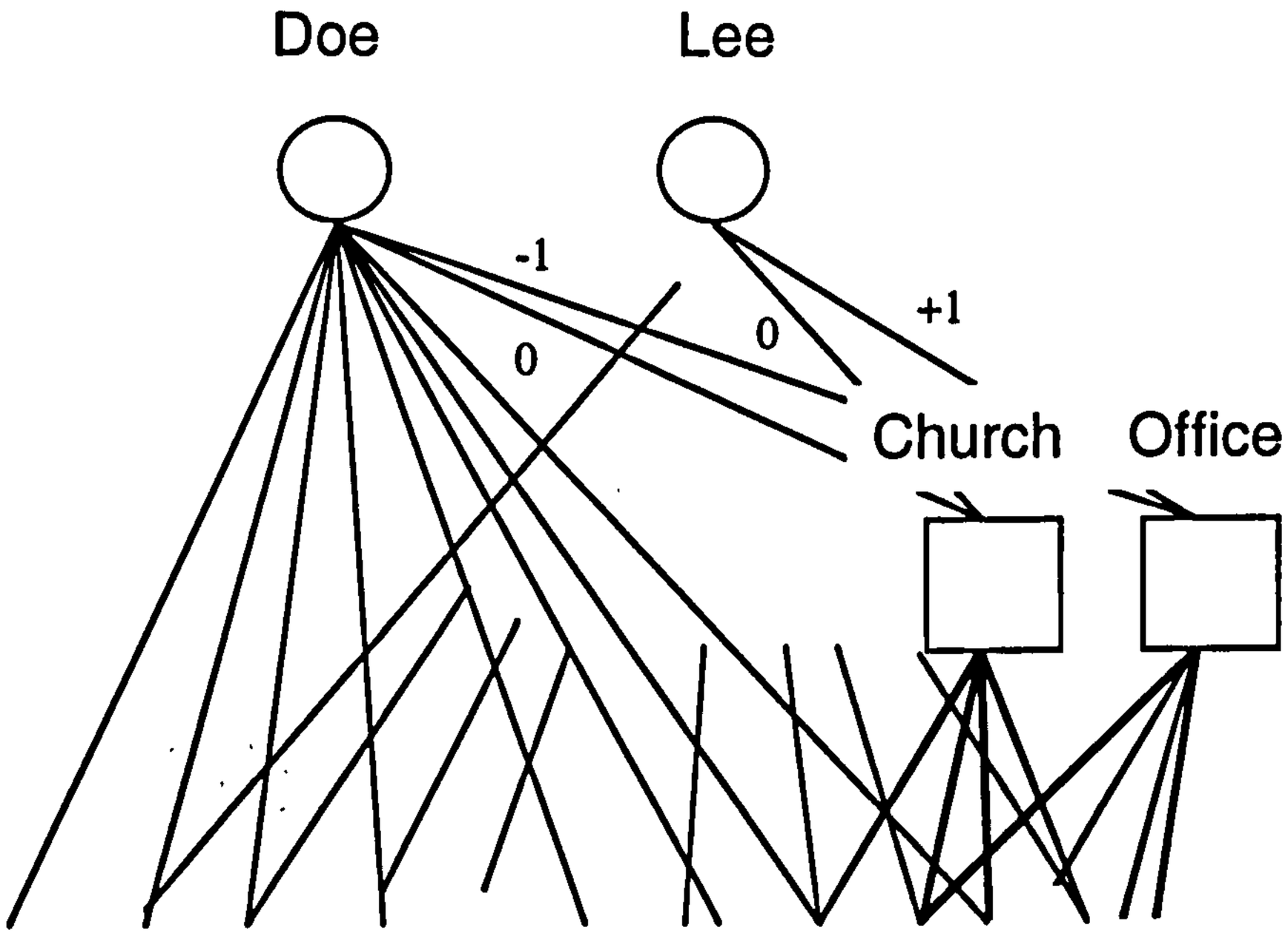


Figure 3.8 Illustration of Baywatch model with strong hint. Weights in bold are fixed connections.

As to be expected, Lee Critical Presented and Unpresented were classified best with the Strong True hint, followed by the network with the Weak True hint, the Weak False hint, and the Strong False hint. Slightly surprisingly, the Filler feature performance was in the reverse order, so that correct classification of presented filler features was best with the Strong False hint, as shown in Figure 3.9. These results can be explained as follows. With the Strong True Hint, whenever a Lee item is presented, there is no error on the output units. Because of the hint and because each training item contains a Critical feature, the weighted sum of the inputs to the category units is +1 and -1, as it should be.

This means that no adjustment takes place on any weight for this trial, or indeed

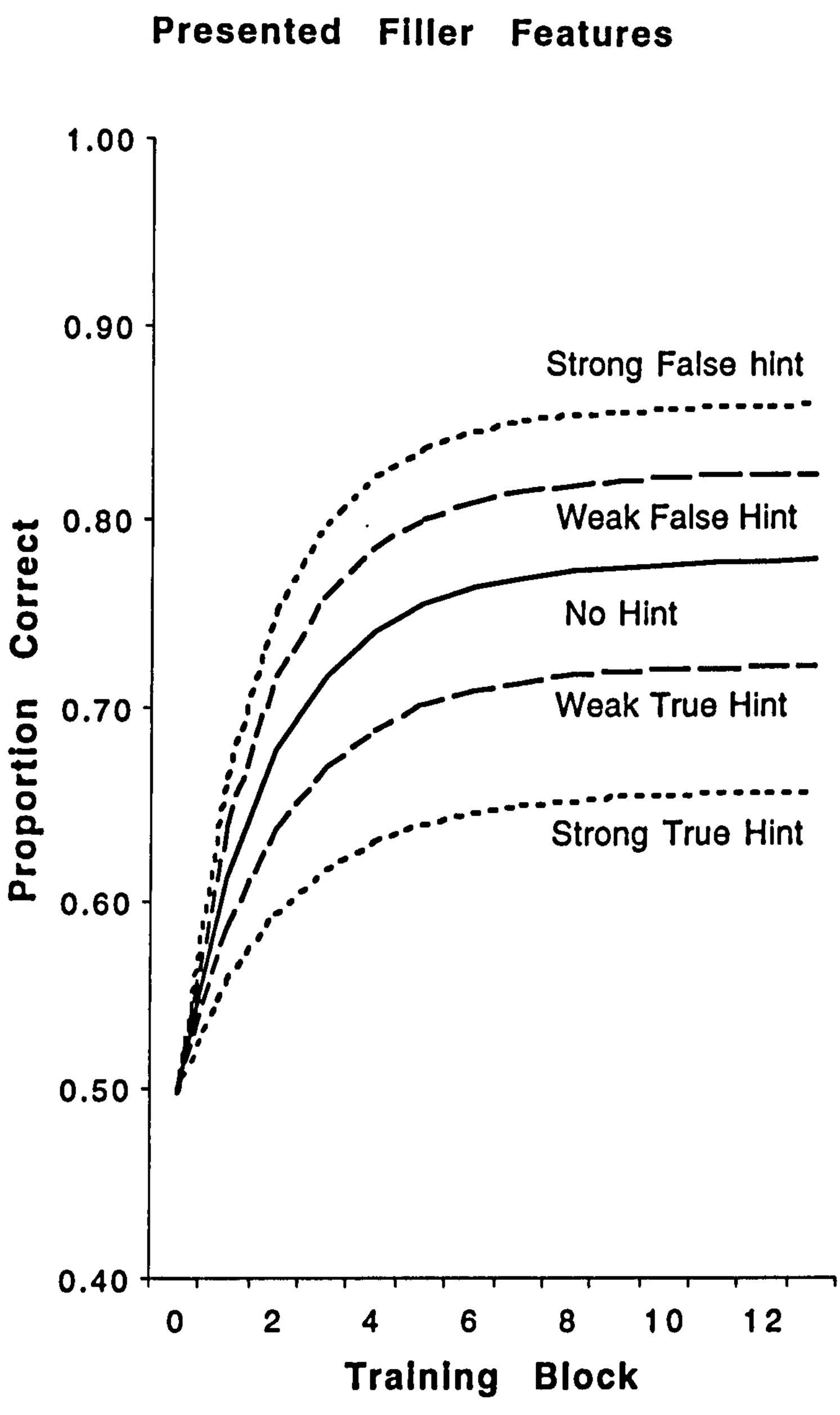


Figure 3.9 Predictions from training flexible, pre-programmed weights.

on any Lee trial. However, learning does take place on Doe trials, because there are errors on the output units. Filler items therefore suffer when True hints are

given to the network, because there are fewer trials in which learning takes place. False hints allow more weight adjustment for the Filler connections and therefore facilitate the learning of filler features, but at the cost of the Critical features.

3.1.4 Evaluation of Baywatch

The principle finding of the Heit and Bott (2000) experiments, that of increased effects of prior knowledge over learning, was replicated by the model. One notable difference between the model's predictions and the participants performance however, is that the model would predict a robust effect of presentation frequency, that is, more accurate judgements for features presented twice per block compared to features presented once per block. In contrast, there was no significant difference between these two levels of presentation in the Heit and Bott (2000) experiments. This insensitivity to frequency could be an important aspect of concept learning in knowledge-rich domains (cf. Murphy & Allopenna, 1994) in which case it would be important to try to capture it in a future version of the Baywatch model. Indeed, by introducing a hidden unit to the empirical side of the network (as in the alternative presentation of the model in Section 3.1.1 and Figure 3.3), no frequency effects would arise because of the lack of individual links between features and output nodes. On the other hand, in the present experiments the lack of sensitivity to presentation frequency could just reflect subjects' reading strategies and might be highly dependent on number of features per presentation and the reading time allowed for each presentation. Therefore further experimental study is required.

As well as modelling the basic experimental results, several other simulations were carried out to examine how the model might scale up. In general, these demonstrated that there were few problems with adding extra category nodes; those that are congruent with the training data enhance learning, while those that aren't have little effect. This would seem to be true for the people as well - if a category seems to map onto a learning task, then it will be applied, but if it doesn't help to distinguish the new data, the known concept will have little effect. Of course, a situation might arise where some noisy data activates an inappropriate category node, thereby exaggerating the effects of the (incorrect) data. Again however, this would be a plausible response by the model - people seem especially susceptible to error which maps onto their background knowledge.

Another set of simulations were used to test some experimental possibilities. The first of these involved presenting data to the model with some incongruent items, that is, items which go against the categories suggested by the other items. This manipulation demonstrated that the incongruent items were learnt worse than congruent and filler features. Second, some 'hints' were given to the model, in terms of non-zero weights on the category to output nodes. One unexpected finding from these simulations is that performance on filler items depreciates as performance on the critical items improves. In other words, there is some overshadowing of the filler items, as observed when the standard network was compared with the knowledge-based network in Section 3.1.2, shown in Figure 3.5. These predictions were investigated empirically and are described below in Section 3.2.

Finally, it is worth pointing out that the Baywatch model as presented in this chapter is but one possible variant within a larger class of models that could perform knowledge selection. For example, referring to Figure 3.2, the model could have category label units (Doe and Lee) added to the input layer as well as feature units (F0, F1, etc.) added to the output layer, turning the model into an auto-associator. Such a model could make a greater variety of inferences, such as feature-to-feature inferences (e.g., Heit, 1992) in addition to the feature-to-category inferences in the present version of the model. Hence the auto-associator version could be applied to a wider range of experimental tasks.

There are several other ways that the architecture of the Baywatch model could be modified. These changes were not necessary for fitting the results of experiments so far, but they could be useful for application to other experimental designs. First, the various modules in the network, including the empirical module and all the PK nodes, could be placed in greater competition with each other. The present architecture of Baywatch encourages co-operation between different modules, in the sense that outputs from multiple modules are combined to make a prediction. Instead, the network could be encouraged to specialise, for example learning that different modules should be used for different stimuli. Some stimuli might be best classified with the empirical module alone, whereas other stimuli would be best classified based on a single PK node. This scheme would force the network, for example, to choose between a church PK node and a cathedral PK node, rather than allowing their influences to combine (see Jacobs *et al.* 1991, for a further discussion of ways to increase competition between

modules). Third, learning could be allowed on the connections between Critical input features and the PK nodes. At present these connections are fixed at the start of learning, but it is possible that allowing these weights to change slowly would allow the network to address the issue of how global theories might change over time. That is, people may have a set of prior concepts that help learning, but these concepts themselves could be modified occasionally.

3.2. Experiments

This section reports the results of two experiments designed to test the predictions of the model, described in Section 3.1.3 above. The first involves testing how knowledge may harm the learning of individual features, while the second examines whether items that are incongruent with the predominant knowledge are more difficult to learn. An additional hypothesis concerning increased presentation time was investigated in the second experiment, following Heit's (1998) finding that slower-paced learning alters the effects of prior knowledge. The general experimental procedure is the same as that described in Heit and Bott (2000), although the frequency manipulation was omitted because of a lack of a reliable difference between conditions.

3.2.1 Experiment 1

When the model was provided with a true 'hint', the relevant critical features were seen to be learnt more quickly. However, performance on filler features suffered. This could be interpreted psychologically as a blocking effect, or as overshadowing (for example, Gluck & Bower, 1988). In the typical blocking paradigm, a two-phase learning design is used. In Phase 1, participants learn to predict an outcome on the basis of a single valid cue. In Phase 2, a second redundant cue is constantly paired with the already established valid cue. What typically happens is that participants are reluctant to predict the outcome on the basis of the second cue alone, even though it is perfectly correlated with the outcome in Phase 2. The effect has been observed extensively in both the animal

and human learning literature, although never in an experiment with a single learning phase. These next experiments can be seen as an attempt to reproduce the blocking effect observed in the model.

The general procedure is as described in Heit and Bott (2000) (and in this Chapter's introduction), but with some exceptions, namely, the hint and the removal of the frequency manipulation. The quantity of prior knowledge was manipulated through the instructions given to participants. Specifically, one set of participants received instructions giving them the hint, for example that Doe buildings were very similar to churches, whereas the other group did not. The hint was true, in the sense that the knowledge agreed with the data they were to be presented with. Critical features were expected to be learnt better in the Hint condition, although Filler features should be better in the No-Hint condition. Finally, it is worth noting that when the simulations were carried out, there were differences between the target category which received the hint (say, "Doe is like a Church"), and the category which didn't (Lee). The situation is not as straightforward in the experiment however, because participants may well logically deduce that *not* being Church means the category must be a Lee.

Method

Participants

Sixty-six University of Warwick students participated and were paid £4 or received course credit. Thirty-three students were randomly assigned to each condition in the Hint / No-Hint factor.

Design and Stimuli

The experiment was divided into a Training phase and a Testing phase. In the Training phase, participants observed a series of exemplars together with the appropriate category label. Exemplars were presented on a computer screen, in a random order. Participants were not required to say which category the exemplars belonged to, merely memorise the information they were given. There were 5 Training-Testing blocks in the experiment. Ten different exemplars were presented in a training block, with a new set of exemplars generated for each block according to the rules described below. Thus, each participant saw a total of 50 distinct exemplars in the experiment. During the Testing phase, participants were asked to say whether individual features were more likely appear in one category or the other.

Participants learned about two categories, "Doe" and "Lee" buildings. The allocation of Doe / Lee labels to church / office categories was made at random for each participant. Exemplars were descriptions of buildings, with one

attribute of the exemplar presented on each line together with the category label above it. Each was shown on a computer screen, one per page, for 6 seconds. Exemplars were constructed of three different types of features: Critical features, which were expected to be influenced by prior knowledge; Filler features, which shouldn't be affected by prior knowledge; and Individuating features, which were simply used to slow participants down and played no part in the design. The features were pre-tested by Heit and Bott (2000) in a free-sorting category construction task. We showed that the Critical features were generally grouped together in the same way, while Filler features were randomly placed into categories. A list of features is shown in Table 3.1.

Each exemplar consisted of two Critical features, two Filler features and three Individuating features. In each block, exemplars were created so that 6 out of 8 pairs of Critical features appeared and 6 out 8 pairs of Filler features. The other two pairs were reserved to be Unpresented features for the Testing phase. These same features remained Unpresented throughout the experiment. All features appeared twice (or not at all) in each block, so, unlike Heit and Bott (2000), there were no frequency manipulations. There were 40 Individuating features and three was chosen at random for each exemplar. Participants were asked about all features during the Testing phase: 8 Critical features with 2 tokens each; 8 Filler features with 2 tokens each; and 40 Individuating features; making a total of 62 questions.

Procedure

Both groups received general instructions telling them that they were going to be learning about two types of buildings, Doe and Lee buildings. In addition, the Hint group received an extra page (screen) saying "Lee [or Doe] buildings are like Churches in a number of ways". The Hint was always for Churches, but the Doe / Lee decision was made on the basis of what the particular training set indicated (which was determined at random).

At the end of each Training and Testing block, participants were presented with a screen informing them that they were about to start the next phase. On completion of the Testing phase, they were provided with a score indicating the percentage of features they got correct. Because this score included the Individuating features (which were very difficult to learn), it was highly variable and unlikely to guide the choice of responses to Unpresented items.

Results

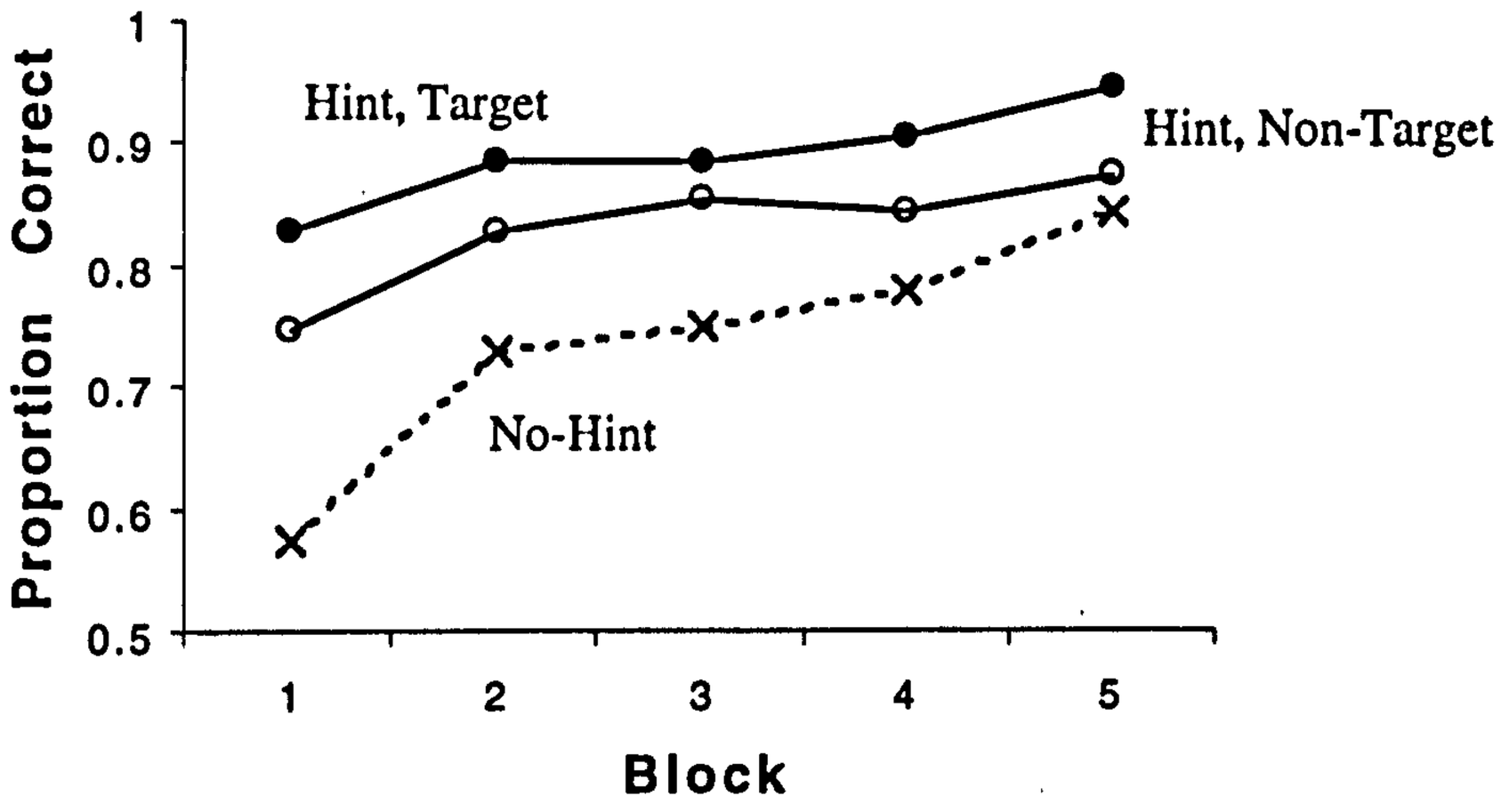
The following within-subject factors were involved in the analysis: Block (5), Feature Type (Critical or Filler), Frequency (Presented or Unpresented) and Target Category (2). Target Category refers to whether or not the category label is directly suggested by the Hint. For example, after being given the Hint, 'Doe buildings are like Churches', the feature 'Is lit by candles' is a Doe feature, and therefore falls into the 'Target' condition of the Target Category. Conversely, the feature 'Is lit by fluorescent lamps' is a Lee feature, and is therefore

described as being from the 'Non-target' condition. Finally, there was a between-subjects Instructions manipulation, either Hint or No-Hint. Note that the Target Category factor only applies to the Hint condition.

The basic knowledge effect observed in Heit and Bott (2000) was replicated. Ignoring the Hint manipulation for the moment, there was a main effect on the Presented features of Type, $F(1,64) = 22.76$, $MSE = 0.09$, $p < 0.0005$ and the interaction of Type by Block, $F(4,256) = 3.1$, $MSE = 0.03$, Hunyh-Feldt Epsilon $= 0.66$, $p = 0.016$, such that Critical features were learnt increasingly better than Fillers as learning progressed. Furthermore, unpresented Critical features were responded to more accurately as the experiment went on, $F(4,256) = 14.23$, $MSE = 0.72$, Hunyh-Feldt Epsilon $= 0.86$, $p < 0.0005$. In other words, the effects of prior knowledge increased throughout learning.

For the Unpresented items, a significant main effect of Hint was observed, $F(1,64) = 6.83$, $MSE = 0.19$, $p = 0.011$, such that those who received the Hint performed better on the Critical features than those who didn't, although there was no interaction with Block ($p = 0.101$). Next, the differences within the Target Category factor were examined. This analysis is only sensible on the Hint group, because the two levels of the Target Category are created by the Hint instructions. For Unpresented features, there was a reliable main effect of Target Category, $F(1,32) = 5.87$, $p = 0.021$, but no interaction with Block ($p = 0.752$). The upper panel of Figure 3.10 shows the Unpresented features as a function of the Hint manipulation and the Target Category.

A similar analysis was carried on the Presented features, but with the added complication of a Filler features as well as Critical ones. First, all factors were subjected to an ANOVA. This failed to yield any reliable effects of the Hint manipulation, with all p's > 0.1. Another analysis was carried out on the Hint group alone, to establish whether any effects of Target Category were present. The main effect of Target Category was narrowly non-significant, with $F(1,32) = 3.23$, $p = 0.08$, but no others involving the Target Category effects. Because of the closeness of this result, and because of findings from the Unpresented features, the differences between the Hint group's Target Critical features and No-Hint group's Critical features (note that there is no Target factor in this group) were subjected to an ANOVA. This revealed a significant main effect of the Hint, such that those receiving the Hint scored more highly than those that didn't, $F(1,64) = 10.03$, $MSE = 0.2$, $p = 0.002$. The lower panel of Figure 3.10 displays this trend and also demonstrates that all five block means for Hint Critical are higher than any of the other conditions ($p = 0.031$ on a Binomial test). In summary, there is small amounts of evidence suggesting that the Critical features benefited from the Hint, but that Filler features were unaffected.



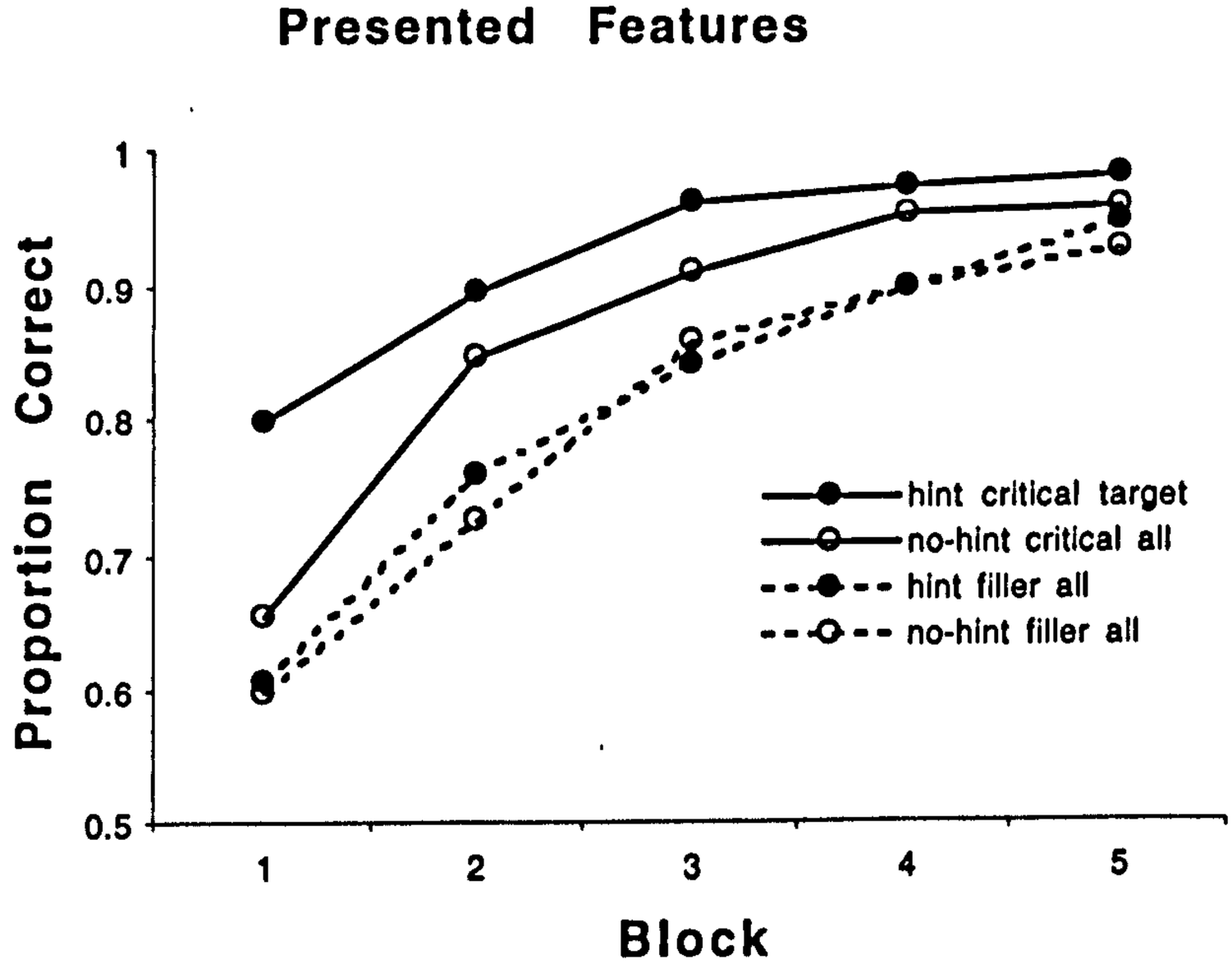


Figure 3.10 Results from Experiment 1.

Discussion

The basic knowledge effects observed in Heit and Bott (2000) were replicated. This meant that there was more of an influence of knowledge as the experiment progressed, as manifested by increasing differences between presented Critical and Filler items, and the deviation of unpresented Critical items from chance. Furthermore, the Hint given to participants produced reliable improvements in accuracy for the unpresented items, as predicted by the model. Contrary to predictions however, there was only slight improvement in accuracy for Critical features and no reduction in accuracy for presented Filler features.

The weakness of the effect on presented Critical features is slightly puzzling, given that it was possible to achieve a perfect score on the basis of the Hint alone. Furthermore, the Hint produced at least a 0.2 increase in accuracy for the Unpresented features, so participants could clearly understand the information given to them. An improvement could be made in future experiments by re-wording the Hint slightly, for example, instead of simply saying "Doe buildings are like Churches", a sentence could be added which reads, "This means that if you see a feature which looks as if it belongs to a Church, then you can assume that it belongs to a Doe building". However, a deeper problem might be that participants do not trust information given to 'help them' in psychology experiments; they may prefer to trust their own memories as far as possible, and only resort to the Hint when they have no idea whatsoever, that is, only for Unpresented features.

Despite the weakness of the effect, it is important to realise that the Hint did cause changes for the Critical features, yet there was no suggestion of a blocking effect on the Filler features. One obvious reason why blocking may not have occurred was that after the first test trial, participants were aware that they would be tested on all the features, including the Fillers. This might have encouraged them to learn the Individuating features as well as the Doe / Lee exemplars. Worse still, those with the Hint may have decided to concentrate their attention on the Filler features, because they feel they can easily identify the Critical ones (however, there is no evidence of this in the data). To answer this criticism, another experiment was run where participants were only tested once, at the end of the five blocks. However, after twenty participants had completed the experiment, there was no evidence for worse behaviour in the filler items: it does not seem that testing between blocks affects the learning strategies of the participants.

Along the same lines, another potential explanation is the lack of interactive learning during the experiment. Most demonstrations of blocking have involved the participants classifying the whole exemplars during the training phase, and afterwards being tested on the individual features. If that design had been used in this experiment, participants might have been content to use only their prior knowledge to classify the training instances and not to learn the Filler features. Furthermore, if they were provided with training examples until they reached a criterion (and no further), this might have encouraged them to only use the features they knew, rather than waste time learning other features.

However, following completion of these experiments, Kaplan and Murphy (2000) reported an extensive study to investigate a very similar idea. Their approach was to use a between subject design with one group receiving a thematically linked category task, while the other group's exemplars did not conform to a known category. The thematic category also contained features which were not linked to the prior knowledge (equivalent to Filler features), and blocking was expected to occur on these. Participants were shown exemplars until they reached a criterion of classifying all exemplars correctly in a single block, followed by a testing phase on the individual features. Contrary to Baywatch's predictions, some of Kaplan and Murphy's experiments demonstrated an *improvement* in performance for Filler features in the thematically linked category, and no blocking effect was observed in any of the five experiments. Kaplan and Murphy suggested that this improvement was because participants were incorporating the Filler features into their prior knowledge as the experiment progressed. Thus, the Filler features would become part of the old category, and be treated as a Critical feature by the end of the experiment. In terms of the model, this could be achieved by incrementing the weights from the Filler features to the PK nodes as learning continues. Blocking would then be reduced, or even removed completely by the end of learning.

It appears that blocking of Filler features is a difficult result to find, whether through providing a 'hint' or through knowledge versus non-knowledge tasks (Kaplan & Murphy, 2000). Certainly, Kaplan and Murphy's (2000) suggestion of incorporating Filler features may explain why blocking did not occur in these

experiments. However, in Chapter 2, the discussion on the "curse of dimensionality" illustrated that not paying attention to some dimensions would be a highly desirable result of applying prior knowledge. Add to this the extensive literature on blocking in general, and one is left with a very strong theoretical case that blocking should occur in some situations. Perhaps future experiments might increase the number of attributes for each exemplar, thus highlighting the advantages of reducing the dimension search.

3.2.2 Experiment 2

The simulation involving the Critical Incongruent feature (Section 3.1.3) suggested that the knowledge effect should be reduced, and that the Incongruent feature should be learnt less well than the Filler features. This experiment tests these predictions. The design mirrors that of the simulation, in the sense that two out of the six Critical features presented during training were selected to be at odds with the Church or Office block category. For example, if the feature "has wooden furniture / has metal furniture" was selected, then "has wooden furniture" would appear with other features suggesting an Office block and "has metal furniture" would appear in the Church category.

The many experiments on prior knowledge on category learning (Hayes & Taplin, 1992; Heit, 1994, 1995, 1998; Heit & Bott, 2000; Murphy & Allopenna, 1994; Murphy & Wisniewski, 1989; Pazzini, 1991; Wattenmaker, Dewey, Murphy, & Medin, 1986; Wisniewski, 1991, 1994) seem to indicate that knowledge will facilitate learning when it is consistent with the category structure, and slow the learning when it is inconsistent. However, only Heit's

studies have examined the situation where knowledge is expected to help some features and harm others: it is much less clear cut whether people will apply their knowledge in these situations.

Heit demonstrated that incongruent items are treated differently in a learning task to those that are congruent. In these experiments, participants saw descriptions of people in an imaginary city, some of which conformed to prior knowledge, such as "shy, and does not go to parties", and some which didn't, such as "shy and goes to parties often". When they were asked to say what proportion of people in the city had these features, the congruent pairings received higher estimates than the incongruent features, despite participants having seen equal quantities of both in the learning phase. It is difficult to draw firm predictions from these however, because Heit asked participants to use both the examples they had just seen, *and their general knowledge of cities.* It is therefore unclear whether the proportion of incongruent features were remembered less accurately, or whether participants were altering their estimates on the basis of what other cities were like.

On an intuitive level, participants might respond in one of two ways to Incongruent features. First, they could perform poorly on these items because of the mis-match between their knowledge and the feature assignment, in the same way that the model predicts. Second, they could notice the incongruency and perform *better* on these items; it is not uncommon to register an item which is the 'odd one out' in a category, and then remember it better than the others.

In addition to testing the simulation predictions, a manipulation of presentation time was introduced. Heit (1998) found that altering the length of time that participants viewed each exemplar changed the effects of background knowledge. In his task, participants saw descriptions of people in an imaginary city, some of which conformed to prior knowledge, and some which didn't (as described above). The extent to which features were incongruent was a repeated measures manipulation, so that the training examples contained either 0%, 25%, 50%, 75% or 100% items incongruent with prior knowledge. The task for the participants was to estimate the conditional probability that another person from a new city, with a characteristic such as being shy, falls into the congruent category, such as not going to parties, or into the incongruent category.

With the fast presentation speed, there was a constant difference between the estimates of the proportion falling into the congruent class, versus the proportion incongruent, across the different percentages. The upper panel of Figure 3.11 shows the results of this experiment, with the constant effect of knowledge clearly shown by the parallel lines. However, at a slower pace of learning, the effects of knowledge are reduced, as shown by the curved lines in the lower panel of Figure 3.11. Heit (1998) hypothesised that this was due to participants choosing to selectively weight the incongruent items, when given enough time to do so. When these results are translated into predictions for this methodology, reduced effects of prior knowledge are to be expected with slower learning. This would be because the incongruent features become more weighted, and the effects discovered in the simulations will therefore become exaggerated.

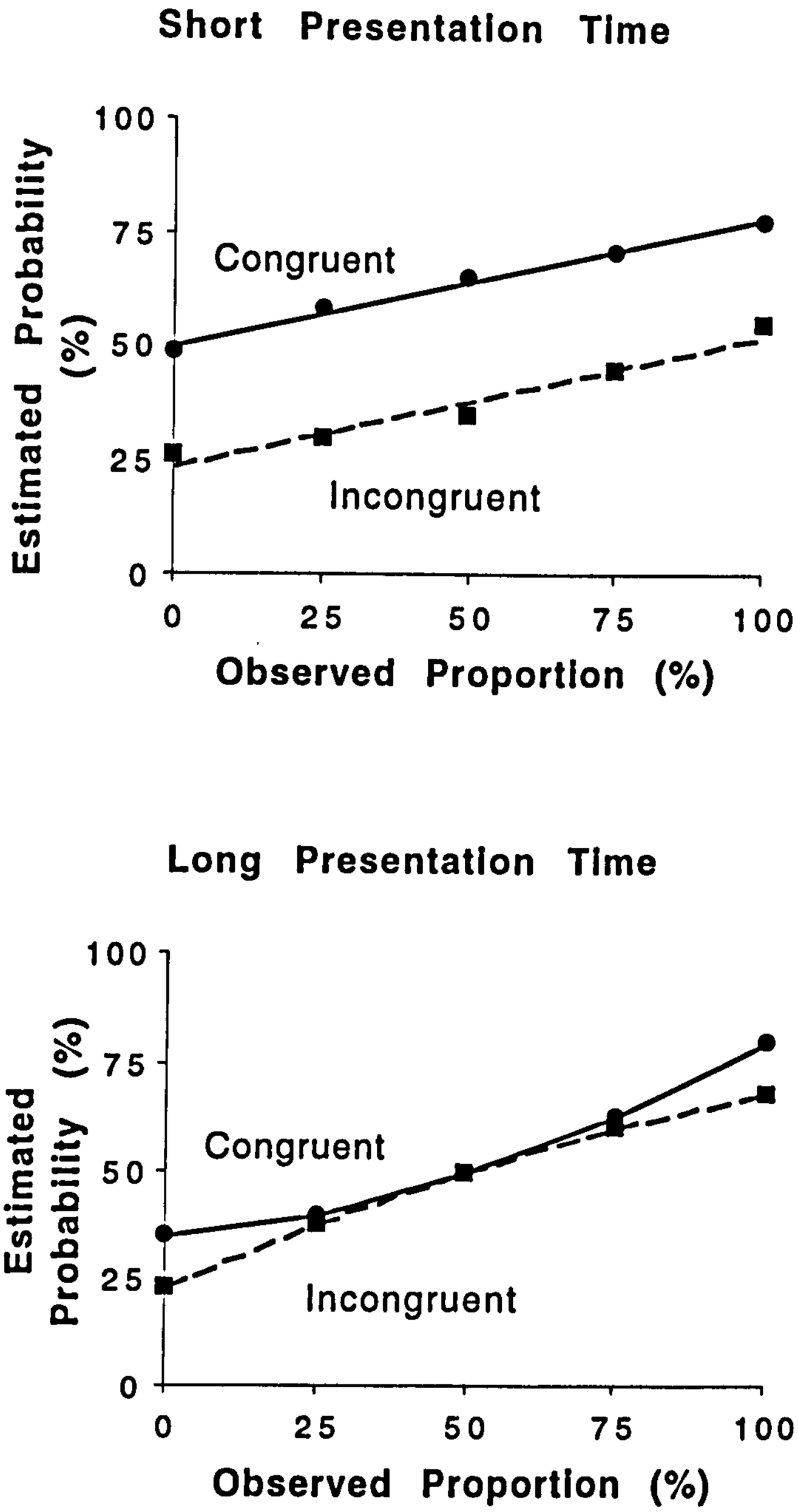


Figure 3.11 Results from Heit (1998), illustrating the effects of length of presentation time on responses.

As in the previous experiment, there are some changes to the basic design in Heit and Bott (2000): again the frequency manipulation was dropped, but also the number of individuating features was reduced; is was felt that the task was slightly more difficult now and this would equate the level of difficulty with that of Heit and Bott.

Method

Participants

Ninety-seven University of Warwick students participated for course-credit or a cash payment of £4. Forty-seven were randomly allocated to the Fast presentation group (see below), and 50 to the Slow presentation group.

Design and Stimuli

The same set of buildings stimuli was used as in Experiment 1. As before, the training set consisting of 6 Presented Critical features and 2 Unpresented. However, to instantiate the congruency manipulation, 2 of the 6 Critical Presented features were randomly selected to be Incongruent for each person (Incongruent Unpresented do not exist, by definition). The Incongruent features remained the same throughout each person's learning phase, that is, there was no variation in the congruency of a feature once the experiment had began. As described above, an Incongruent feature was one whose feature value went

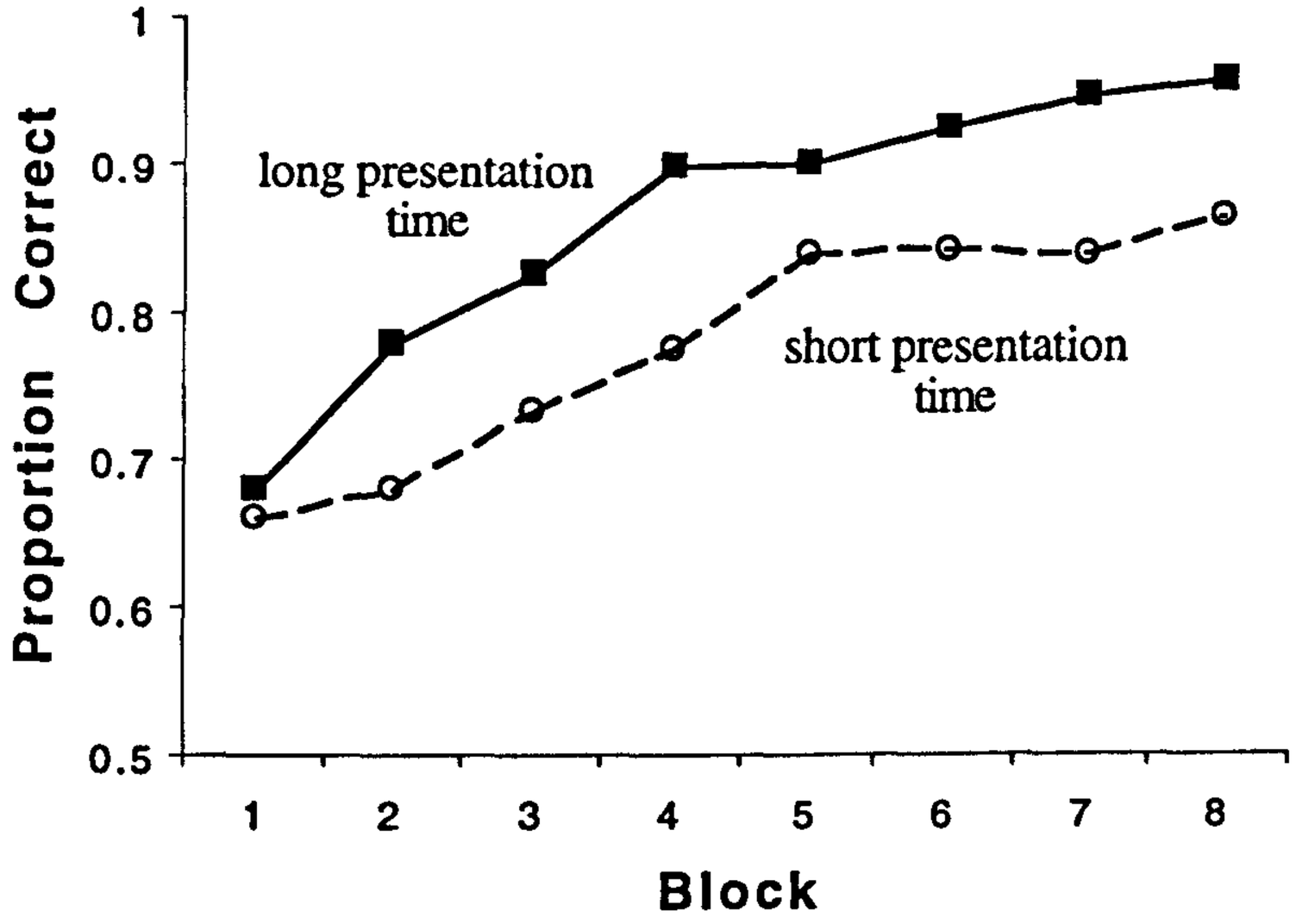
against the prevailing Church / Office block mapping. Allocation of Filler features was identical to Experiment 1. In summary, there were now three types of presented feature: Critical Congruent; Critical Incongruent; Filler; and two types of Unpresented feature: Critical and Filler (although the Fillers must be at chance, by definition). There was no presentation frequency manipulation - all features appeared twice per block, or not at all. To make the task slightly easier for participants, the number of individuating features was reduced from 4 per exemplar to 2 per exemplar, and they received eight blocks of learning trials.

There was also a between-subjects manipulation of Presentation time, such that one group received 15 seconds to view each exemplar (the Slow group), while the other group were allowed 6 seconds (the Fast group). All other aspects of the experiment were identical to Experiment 1, No-Hint condition.

Results

For the presented items, the main effect of Presentation Time was significant, such that participants who received more time performed more accurately, $F(1,94) = 16.02$, $MSE = 0.3$, $p<0.0005$. The effect of Block was also significant, with participants getting more accurate as the experiment progressed, $F(7,658) = 46.87$, Huynh-Feldt Epsilon = 0.83, $MSE = 0.05$, $p<0.0005$. The upper panel of Figure 3.12 displays accuracy as a function of block and presentation time. Surprisingly, there was no main effects or interactions involving Feature type (p's > 0.3), contrary to Heit and Bott's (2000) observed differences between

Critical and Filler features. Because asymptote was reached early in the experiment



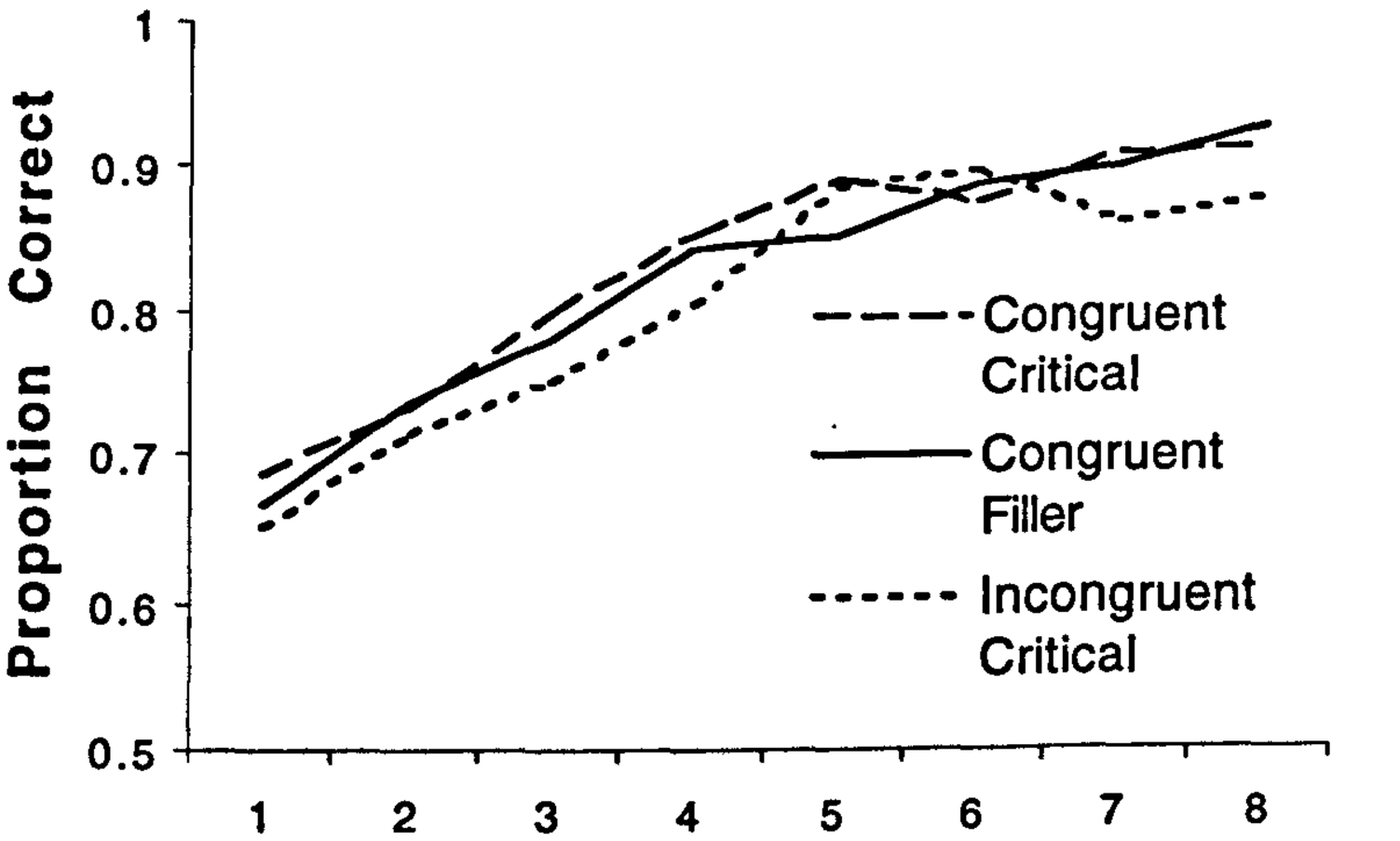


Figure 3.12 Results from experiment 2.

(Block 4), another ANOVA was conducted on just the first four blocks, but again with no interesting effects emerging. Finally, the lower panel of Figure 3.12 indicates that all but one of the Incongruent Block averages are lower than the Critical averages, which is significant on a Binomial test with $p = 0.035$ (although the Incongruent versus Filler features is not significant, $p = 0.145$). This indicates a very slight effect of feature type.

For the unpresented features, there was no effect of Presentation Time and no effects of Block (p's > 0.25). Because performance on unpresented Filler features was 50 % by definition, Critical accuracy was subjected to a 1-sample t-test with mean 0.5. After aggregating over Block and Presentation Time, this yielded a reliable deviation from chance ($t(95) = 2.59$, $p = 0.011$). Average accuracy was 0.57.

Discussion

There was only a very slight indication of differences between the different feature types on the presented items. The effects observed in Heit and Bott (2000) were not found, and the only evidence of a difference between the Critical Incongruent and Congruent features was in the ranking of the feature block means: the Incongruent means were reliably below those of the Congruent means. However, the effects of prior knowledge were apparent in the large deviation from chance of the Unpresented Critical features, although there was no interaction with block. The presentation time manipulation produced slightly

more accurate responses for the longer presentation time, but no interactions with feature type, contrary to predictions.

These results raise the question of what happened to the knowledge effect in general: why was it so much weaker here than in Heit and Bott (2000), or in Experiment 1? There were two differences between this and the previous experiments. The first is that there were fewer individuating features. This probably made the task easier and explains why performance after Block 1 was noticeably higher than in Experiment 1. However, it is unlikely that a ceiling effect destroyed the knowledge results because participants did not reach their asymptote before Block 5 (and an ANOVA restricted to Blocks 1-4 failed to show differences between features). Further evidence against this is provided by the relatively low Unpresented Critical accuracy (mean of 0.6 at Block 8), as compared to the score in Experiment 1 (mean 0.8 at Block 5); if the knowledge effect were simply obscured by the ease of the task, then there would be no reason to expect this discrepancy between the two Experiments.

The other difference between the designs was the introduction of the Incongruent features. These could have lead to the reduced knowledge effects in two ways. First, there could be less of an association between prior knowledge and the target categories, caused by the Incongruent items adjusting the connections in the 'wrong' direction (as the model simulations demonstrate). Although the simulation showed that knowledge effects should still be preserved, they could have been weakened to such an extent that they became unobservable in an experimental context. The second possibility is that participants are performing

an explicit, conscious search of the hypotheses relating the data to known concepts and that the Incongruent features mean that the church / office block hypothesis is eliminated. For instance, the idea that one of the buildings might be an office block would be disconfirmed on seeing the feature 'Is lit by candles'. Clearly, it is impossible to tell which reasons apply in this experiment, but there are at least hints about what might have taken place from a study carried out by Murphy and Kaplan (2000), published after the completion of these experiments.

Their study involved testing whether there were interactions with prior knowledge and category structure. Prior knowledge was manipulated by having one group learn exemplars which were thematically related, while the exemplars in the other group weren't (in the same way as Kaplan and Murphy, 2000). The category structure could either be 'Factorial', or 'Atypical'. Both structures involve two categories with 6 exemplars in each, described on 5 dimensions, as shown in Table 3.3.

Factorial Structure

		Dimension				
Category	**Exemplar**	**D1**	**D2**	**D3**	**D4**	**D5**
A	A1	1	1	1	1	1
	A2	0	1	1	1	1
	A3	1	0	1	1	1
	A4	1	1	0	1	1
	A5	1	1	1	0	1
	A6	1	1	1	1	0
B	B1	0	0	0	0	0
	B2	1	0	0	0	0
	B3	0	1	0	0	0
	B4	0	0	1	0	0
	B5	0	0	0	1	0
	B6	0	0	0	0	1

Atypical exemplar Structure

Category	**Exemplar**	**D1**	**D2**	**D3**	**D4**	**D5**
A	A1	1	1	1	1	1
	A2	1	1	1	1	1
	A3	1	1	1	1	1
	A4	1	1	1	1	1
	A5	0	0	1	1	1
	A6	1	1	0	0	0
B	B1	0	0	0	0	0
	B2	0	0	0	0	0
	B3	0	0	0	0	0
	B4	0	0	0	0	0
	B5	0	1	1	1	0
	B6	1	0	0	0	1

Table 3.3 Abstract structures of the categories used in Murphy and Kaplan (2000), Experiment 1.

In the Factorial structure, each exemplar has most dimensions with the value typical for its category, but one dimension showing the other value. In the Atypical structure, most exemplars are the prototype, but Exemplars 5 and 6 are highly atypical with several 'crossovers' in each. Murphy and Kaplan (2000) demonstrated that, when using the Factorial structure, learning the thematically related category had a large facilitative effect on performance compared with the non-thematic category. This is an important result because it shows that using features values from the opposing category does not destroy the knowledge advantage, as they did in this experiment. Furthermore, they showed there was *no* knowledge advantage for the Atypical structure group.

Kaplan and Murphy (2000) explained this as follows. In the Thematic Factorial group, participants were able to classify each instance on the basis of whether or not it was an example of their prior knowledge concept, say a Church building, and then make the appropriate response by using the inference "Church buildings are Does". On the other hand, in the Thematic Atypical group, participants could not use this strategy: classifying instance 6 (and possibly 5) as a Church or Office block by, say, a feature count, would produce the 'wrong' answer and consequently be placed in the wrong category. The use of particular prior knowledge concepts would then be abandoned, and some other strategy adopted. This is very similar to the hypothesis searching explanation given above for the current experiment, the only difference being that Kaplan and Murphy tested performance on whole exemplars, not the individual features. In both cases, participants are assumed to apply their knowledge only if it appears to work for the tested items, which is not the case when training with Incongruent features or

Incongruent exemplars (Exemplars 5 and 6 from Table 3.3). This explanation is necessarily post-hoc, and more research is needed before a firm conclusion can be reached as to whether some high-level, hypothesis testing strategy is assumed, or a more associationist, Baywatch-like account is best. Regardless of which of these is appropriate however, a lack of a knowledge effect explains why there was no interesting effect of the presentation time manipulation – with only a weak distinction between Critical and Filler features, differing knowledge effects across presentation times were unlikely to be observed.

3.3 General Discussion

In modelling the Heit and Bott (2000) results, the approach taken here was to assume that there were many expert modules, or hypotheses, that the participants brought to bear on the task at hand. These experts corresponded to known categories, such as a Church or Office Block, and were activated when features belonging to these concepts appeared in the target category. This resulted in more learning on these critical features and a mapping between the target concept and the known categories, thereby reproducing the findings shown in Figure 3.1.

The modelling approach that was used here was the mixture of experts architecture, developed by Jacobs *et al.* (1991) and discussed in the previous chapter. One application that was mentioned was Erickson and Kruschke's (1998) model of how 'rules' and 'similarity'-based categorisation systems interact. By demonstrating that both prior knowledge results and 'rules' can be modelled in the same way, the links between these two areas have been emphasised. In fact, the operational definition of a known concept in this paper has been little more than a semantic version of the perceptual rules that Erickson and Kruschke have suggested participants search for in a standard categorization task. For example, both the perceptual rules and known concepts have been instantiated as relatively inflexible boundaries which participants apply if they work, and abandon if they don't (see Experiment 2 Discussion). The exact overlap between the two dichotomies remains to be seen, but it may worth

considering the relationship between prior knowledge and empirical learning as analogous to that between similarity and rules.

Further simulations demonstrated that the more knowledge the network was provided with, the worse it performed on the filler features, that is, those features which weren't connected to its knowledge already. This is a key prediction of such an error correction algorithm – learning only takes place when the exemplars are misclassified; more knowledge means reaching criterion quicker and consequently fewer learning trials. Environmentally this seems a sensible strategy too: if the organism can find a quick mapping of the desired concept, why bother wasting precious recourses on learning idiosyncrasies? The danger though, is that the only concepts which are learnable are the ones known already. Once again, the implications of the bias / variance dilemma become apparent: by not paying attention to information which doesn't fit in with our background knowledge, the organism runs the risk of missing the concept altogether, or at least failing to acquiré information which might be useful at a later time. Perhaps this is why an empirical demonstration of prior knowledge blocking is difficult to come by: Experiment 1 failed to show the expected deterioration on Filler performance, as did several experiments in Kaplan and Murphy (2000). One interpretation of these experiments is that there is some higher learning rule which encourages us to relate new information to prior knowledge, over and above the immediate statistical advantages. The model presented here can be augmented in ways to account for this and future work will produce predictions. However, it is worth emphasising that blocking effects would still be predicted for some situations, and further experiments are required to find out where.

Another simulation revealed how the model would cope with features which are incongruent with the expected prior knowledge categories. This showed that not only should these features be learnt worse than Congruent ones, but that they should be classified less accurately than the filler features. Experiment 2 investigated this idea, with the finding that the Incongruent features destroyed the basic knowledge effects. The most likely explanation for this is that participants were engaged in active hypothesis searching, discretely confirming or disconfirming hypotheses about what the two target categories might be. Having a feature which appeared to fit into the opposing category eliminated the Church / Office Block hypothesis from the set. If this is the explanation, and further work is needed to confirm this, then it calls into question the process Baywatch uses to model the knowledge effects in the first place: a gradual, weight adjustment approach is not suitable for modelling conscious, hypothesis testing.

Finally, one criticism of the modular approach could be that, by incorporating all 'relevant' categories as modules, the question of knowledge *selection* is being avoided. Clearly, the "frame" problem has not been solved here, but using systems where multiple categories interact with each other is at least a step in the right direction: most organisms will go through a learning situation where they know roughly what knowledge to apply, and it is at this stage where the mixture of experts approach is useful.

3.4 Conclusions

The purpose of this chapter was to simulate how prior knowledge might be selected and used in a category learning task. This was achieved by treating the selection process as a situation where the organism has to estimate the likelihood that one of its known categories generated the data. The more evidence there was of a particular category, the more this category's outcome influenced the final classification. In this way, only mappings which have some predictive power are incorporated into the learning task. This approach succeeded in simulating the effects observed in Heit and Bott (2000), and generated several novel predictions. These included: (1) the more knowledge given to participants, the worse they should perform on the neutral features of the exemplars, and (2) features which were incongruent with known categories should be more difficult to learn than neutral features. Although the two experiments carried out here failed to confirm the predictions, arguments were made in favour of continuing the search for these effects. To conclude, the model provides an excellent starting point for generating ideas for future research on prior knowledge and categorisation.

Chapter 4

The previous chapter examined what the effects of prior knowledge are on category formation. Similar ideas are investigated in the next two chapters, but instead of concentrating on categorisation, the research will focus on how we form mappings from one continuous dimension to another. The reason for this shift of emphasis is partly to encourage the exchange of ideas between these two related fields, and partly to examine the effects of prior knowledge in continuous perceptual domains.

Many physical skills involve learning, or relearning, continuous mappings. For example, given an image of an object on our retina, we can map the values on these spatial dimensions to values on proprioceptive dimensions, and point accurately to that object with our finger (Bedford, 1989). A host of more everyday tasks such as throwing, balancing, judging speed, holding objects, decision making and probability judgements all require knowledge of mapping functions. As a way of examining this phenomenon, researchers have focused on the question of which functions are more difficult to learn than others and what kinds of representations underlie the learning of these mappings (e.g. Brehmer 1974; Carroll, 1963; Delosh, McDaniel, & Busemeyer, 1997; Koh & Meyer, 1991).

The research on representation has focused on whether function learning is achieved parametrically or non-parametrically. Parametric accounts (Brehmer, 1974; Caroll, 1973; Koh & Meyer, 1991; Sniezek and Naylor, 1978) assume that

a suitable function is chosen at the beginning of learning and the parameters optimised from the training data. The function is usually a linear combination of basis functions:

$$y(t) = b_0 f_0[(t)] + b_1 f_1[x(t)] + \ldots + b_k f_k[x(t)] \qquad (1)$$

with the most common choice for the basis functions being $f_k(x) = x^k$. For example, Brehmer assumed that participants chose a cubic polynomial basis and then optimised the coefficients first for a linear function, then a quadratic, then a cubic. Parametric models are defined by the fact that they have a restricted range of allowable solutions with which to fit the data. In Brehmer's model for instance, solutions of order greater than a cubic are assumed to be unavailable. In contrast, non-parametric models (Byun, 1995; Busemeyer, Byun, Delosh, and McDaniel, 1997; Delosh, Busemeyer, & McDaniel, 1997) assume one basis function per data point, and can therefore approximate any solution in the limit. The basis function typically used in these models is a Gaussian function of the distance between the training stimuli and the test item.

Evidence in favour of the parametric models has come from studies indicating that participants find it easier to learn functionality related examples over random examples, and certain functions over others (e.g. Brehmer, 1974; Carroll, 1963; Koh & Meyer, 1991; Brehmer, Kuylenstierna, & Liljergen, 1974; Naylor & Clark, 1968; Naylor & Domine, 1981; Sniezek & Naylor, 1978). For example, Caroll provided participants with pairs of input-output examples of line lengths and asked them to learn these. Some participants were given input lines

randomly paired with output lines, whereas some were given examples which were generated by either linear functions or quadratics. Caroll found that those in the linear condition made fewest errors, followed by those who learnt the quadratic function, and finally those who received the randomly combined pairs. He concluded that because they found it easier to learn the functionally combined examples, participants must have been attempting to fit abstract polynomials to the data. Furthermore, on testing values where participants were required to interpolate, responses were as accurate as those to training values (see Koh & Meyer and Delosh *et al.* for similar results). This was taken as evidence that participants had abstracted beyond the specific training values and formed some kind of functional representation.

There have been very few attempts to explain these findings with a non-parametric account of function learning. Among the first were Busemeyer, Byun, Delosh, and McDaniel (1997), who suggested an exemplar-based, neural network model similar to Kruschke's (1992) model of categorisation. Their model reproduced the order of acquisition effects by assuming the participants start off with certain initial weight configurations. These weight biases encourage some solutions to be found before others when combined with a gradient descent learning algorithm. The inclusion of a generalisation parameter allowed the model to interpolate appropriately.

However, evidence against both strictly parametric and non-parametric models was provided by Delosh *et al.* (1997). They argued that in all experiments which had examined the extrapolation behaviour of participants (Caroll, 1963; Delosh

et al., 1997; Waganaar & Sagaria, 1975), the pattern of responses tended to be linear in the direction of the training function. For example, Waganaar and Sagaria found that when participants were asked to extrapolate from an exponential training set, they consistently underestimated the exponential in extrapolation. Figure 4.1 illustrates the performance of participants and the exponential function which generated the training data.

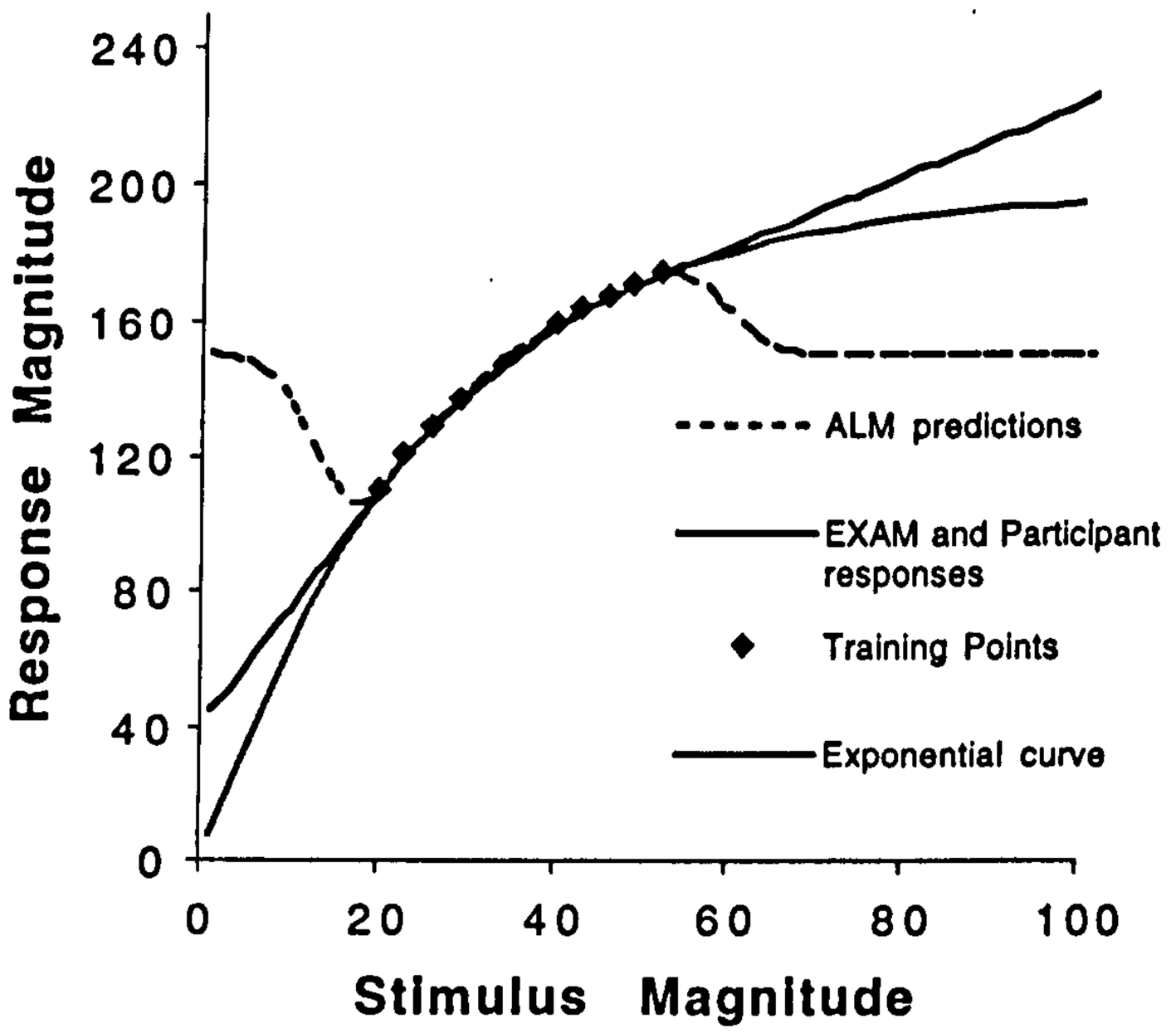


Figure 4.1 Participants' and models' responses to stimuli generated from an exponential curve (Waganaar & Sagaria, 1975).

Note that in the extrapolation regions, the participants extrapolated linearly with the parameters approximately determined by the centre of the exponential and the most extreme training points. As we shall see below, non-parametric models assume the response by the participant is a decreasing function of the distance of the test stimulus away from the training data. This implies that output magnitude should decrease as the extrapolation distance increases – predictions which are

contrary to empirical findings (see Anderson & Finchman, 1996, and Erickson & Kruschke, 1999; for similar arguments concerning participants' extrapolation responses in the categorisation literature).

These findings led Delosh *et al.* (1997) to propose a hybrid model called the extrapolation-association model (EXAM), consisting of a non-parametric representation with a linear extrapolation response rule. They demonstrated that this model fitted the data better than those of Brehmer (1974) or Koh and Meyer (1991), or the straight associative learning model developed in Busemeyer *et al.* The claim of this chapter is that although EXAM performs well in the situations tested so far, it is lacking in several respects. Before these are covered however, EXAM will be described in detail.

4.1 The EXAM Model

EXAM is provided with a number of input values together with appropriate output values. From these, it generates a mapping from a continuous input domain to a continuous output domain. EXAM is best described as being made up of two components. First, some input values become associated with known output values through a training procedure. This process is very similar to Kruschke's ALCOVE (1992). Secondly, an extrapolation mechanism produces generalisation responses based on the output of the first mechanism. It is possible to treat the first mechanism alone as a model of function learning, in which case it is known as the Associative Learning Mechanism (ALM,

Busemeyer *et al.*, 1997). EXAM (Delosh *et al.*, 1997) consists of the ALM together with an additional extrapolation rule.

The ALM has only been defined for a single input dimension and a single output dimension. The input dimension is represented as a set of M input nodes $[X_1, X_2, \ldots, X_i, \ldots, X_M]$, and the output dimension as L output nodes $[Y_1, Y_2, \ldots, Y_i, \ldots, Y_L]$. Each node represents a quantity on a real number line and the total number of nodes reflects the accuracy of the perceptual system. For example, consider a visual system which can see up to length 100cm horizontally. If the system can represent lengths of 1cm, then the input layer for the ALM might consist of 100 nodes. Each input node would respond maximally to a different length of bar so, if a rod of 55cm was being represented, then the node coding for 55 would activate maximally. Note that the nodes are not binary; they can take any value between 0 and 1.

When a stimulus, X, is presented, all the input nodes are activated according to a Gaussian function of the distance between the stimulus and the input node:

$$a_i(X) = \exp\{-\lambda \cdot [X - X_i]^2\} \qquad (2)$$

where a_i represents the activation on input node i and λ is the smoothing parameter (discussed in detail in Chapter 2).

Weights connect the input nodes to the output nodes such that the activation on an output node o_j, is the weighted sum of the activation on the input nodes:

$$o_j(X) = \sum_{i=1} w_{ji} . a_i(X). \qquad (3)$$

where w_{ji} is the weight connecting the input node j to the output node i. The probability that the model generates a particular response Y_j for a given activation on its input nodes, X, is given by the ratio of each output response o_j to the sum of the activation on all of the output nodes:

$$P[Y_j \mid X] = o_j(X) / \sum_{k=1,L} o_k(X). \qquad (4)$$

Note that the probability of generating Y_j is not the same as the output value that Y_j would produce. So, for example, the model might produce an output of 20 when given an input of 2, 20% of the time.

The expected response[1] from the model is the sum of the output values weighted by the probability of generating them. To aid the exposition in later sections, the response is referred to as the mean response to X:

$$m(X) = \sum_{j=1,L} Y_j \cdot P[Y_j \mid X]. \qquad (5)$$

For example, consider a network with just 3 output nodes taking an input value of 2, say. If the activation on one output node was 20, the other 30, and the third 50, then their associated probabilities would be 0.2; 0.3 and 0.5 (from Equation

[1] It is unclear in the original paper whether the ALM produces a single, stochastic response to a stimulus magnitude with an expected (long running average) value across large numbers of trials given by Equation 5, or whether it produces a mean value (a new response) for each stimulus value. For generating quantitative model predictions from the ALM, Delosh *et al.* (1997) assumed the latter instantiation, which is the approach adopted here.

4). The expected response of the ALM would be $(0.2 \times 20) + (0.3 \times 30) + (0.5 \times 50)$, which equals 39.

Learning to associate an input value with an output value is achieved by adjusting the weights between the input and the output nodes. This is achieved as follows. The feedback signal Z, activates the output nodes according to the Gaussian function:

$$f_j(Z) = \exp\{-\lambda \cdot [Z - Y_j]^2\} \qquad (6)$$

Where $f_j(Z)$ is the activation of output node Y_j by the feedback signal Z. The Delta learning rule is used to optimise the weights:

$$w_{ji}(t+1) = w_{ji}(t) + l_r \cdot \{f_j[Z(t)] - o_j[X(t)]\} \cdot a_i[X(t)]. \qquad (7)$$

Where $w_{ji}(t)$ is the weight at time t, $w_{ji}(t+1)$ is the weight at time $t+1$, and l_l is the learning rate.

So far, only the mechanisms of the ALM has been described. The model is perfectly capable of producing continuous mappings, but generates unlikely responses at large distances away from the training data. For example, Figure 4.1 displays the model's responses using a training set generated from an exponential curve. Note that at long distances away from the training data, the model's responses approach the mean of the data set. The reason for this is that as the test values move further away, the differences between the activation

values of the training examples get smaller (because of the flattening of the negative exponential function, see Equation 2). This in turn means that the probability distribution of generating the output values (Equation 4) becomes uniform and hence the ALM's response is simply the arithmetic mean of the training value outputs, regardless of the positioning of the test value (see Equation 5).

Delosh *et al.* (1997) proposed the EXAM model as a way avoiding the unlikely extrapolation patterns shown by the ALM, but keeping the basic representational architecture. They achieved this by adding a linear extrapolation rule onto the responses from the ALM to produce a generalisation pattern which is in the direction of the training function.

The first part of EXAM's extrapolation process involves matching the incoming test value to a training value for which it knows the appropriate output response. The probability with which EXAM matches a stored input value to the test stimulus is given by:

$$P[X_i \mid X] = a_i(X) / \sum_{k=1,M} a_k \qquad (8)$$

This means that the closest input node to the test stimulus will have the highest probability of being selected, then the next closest etc. Once an input node is chosen, three output values are retrieved using the ALM: the response from the chosen node, *m(X)*, and the responses from the two input nodes on either side of the chosen node, *m(X-1)* and *m(X+1)*. These values are then combined to

produce a linear function relating the response to the test stimulus and the distance the test stimulus lies from the selected input node. This linear function is centred on *m(X)*, referred to as the *anchor*, and the gradient is determined by *m(X-1)* and *m(X+1)*. When the stored input node X_i is selected, the expected output response to the test stimulus is given by:

$$E[Y \mid X_i] = m(X_i) + \{[m(X_{i+1}) - m(X_{i-1})]/[X_{i+1} - X_{i-1}]\} \cdot [X - X_i]. \qquad (9)$$

Because the stored input node is selected probabilistically, the expectation must also be taken over all the stored input nodes. Thus, the mean response to a test stimulus is described by:

$$E[Y \mid X] = \sum_{i=1,M} \Pr[X_i \mid X] \cdot E[Y \mid X_i] \qquad (10)$$

In the other words, the mean response is the sum of output values produced when the X_i's are chosen to be the anchors, weighted by the probability that they will be selected in the first place. This implies that the training values closest to the test stimulus have the greatest influence on the extrapolation responses. If the smallest input node, X_1, is chosen, then X_{I-1} in Equation 9 is replaced by X_1, and if the largest input node, X_M is selected, then X_{I+1} is replaced by X_M.

Because this chapter concerns EXAM's extrapolation behaviour, we will go into this in more detail. Consider the points in Figure 4.2. The diamonds linked by a solid line are training points which have been learnt to a high degree (i.e. if the model is tested on one of these, then the output produced by the model will be

approximately equal to the target value). The dotted lines extending out from the diamonds are the results of using each point as the anchor in Equation 9, as a function of the test stimulus cue magnitude. The overall response of the model is given by the sum of output from each of these functions, weighted by the probabilities that each of the training points will be selected as the anchor (Equation 10). For example, consider the case where the test stimulus is at X = 80. For simplicity, assume that only the last three training points have enough probability of being selected as the anchor to influence the final response (the others are too far away). The three remaining training points are selected with probabilities 0.5, 0.3 and 0.2, reflecting their distances from X = 80. The three relevant dotted lines are the bottom three at X = 80, producing the values -100, -180, and -200 from the top down. This means that the final response is (0.5 × -100) + (0.3 × -180) + (0.2 × -200), which equals –150.

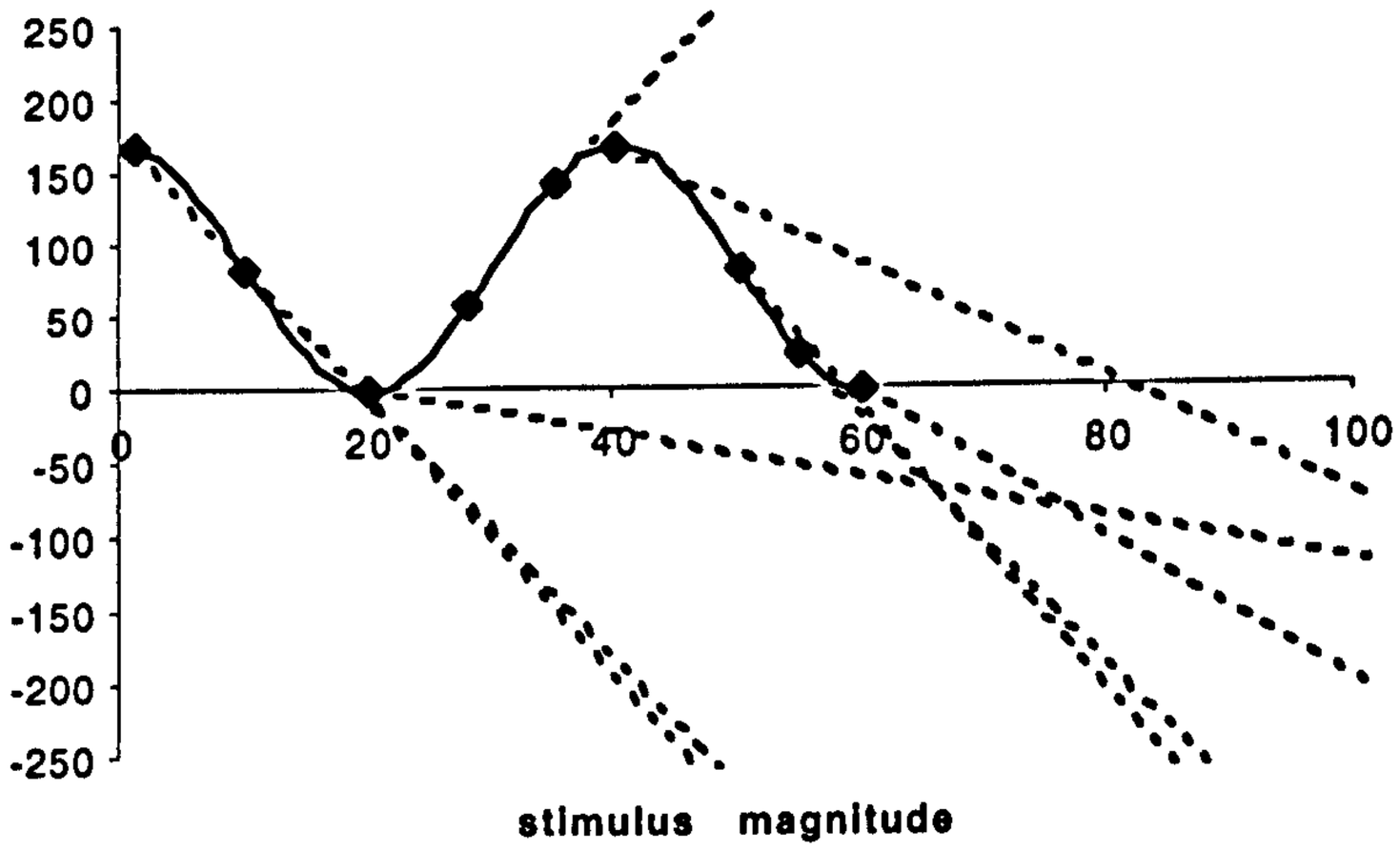


Figure 4.2 Extrapolation mechanism of EXAM. Diamonds correspond to learnt data points. Dotted lines are a particular training point's extrapolation pattern.

EXAM's response mechanism is an ingenious algorithm for generating linear extrapolation from any set of training data. There are several potential problems with it however, which this chapter addresses. First, it does not assume linear interpolation. Experiment 1 tests the possibility that generalisation between two training points follows a linear pattern. The experiment also acts as a pilot study to investigate the methodology used in this chapter and the next. Secondly, an extrapolation mechanism which can *only* linearly extrapolate may not be flexible enough to account for behaviour within the paradigm developed by Delosh *et al.* (1997). Experiments 2 and 3 examine this issue. Finally, an alternative model to EXAM is presented and fitted to the data.

4.2 Experiments

Experiments in this chapter involved participants learning input-output examples and then being asked to interpolate and extrapolate from these. Participants went through training phases, when they received feedback on their responses, and test phases, where they received no feedback. Stimuli were presented in the form of horizontal bars, as shown in Figure 4.3. The length of the bottom bar corresponds to the magnitude of the input, the middle bar is used by participants to enter their responses and the upper bar is used to provide feedback where necessary. Because the goal of these experiments is to test EXAM's predictions, the presentation format of most of the experiments was designed to be identical to those of Delosh *et al.* (1997). Where departures from the methodology are made, explanations are given as to the reason.

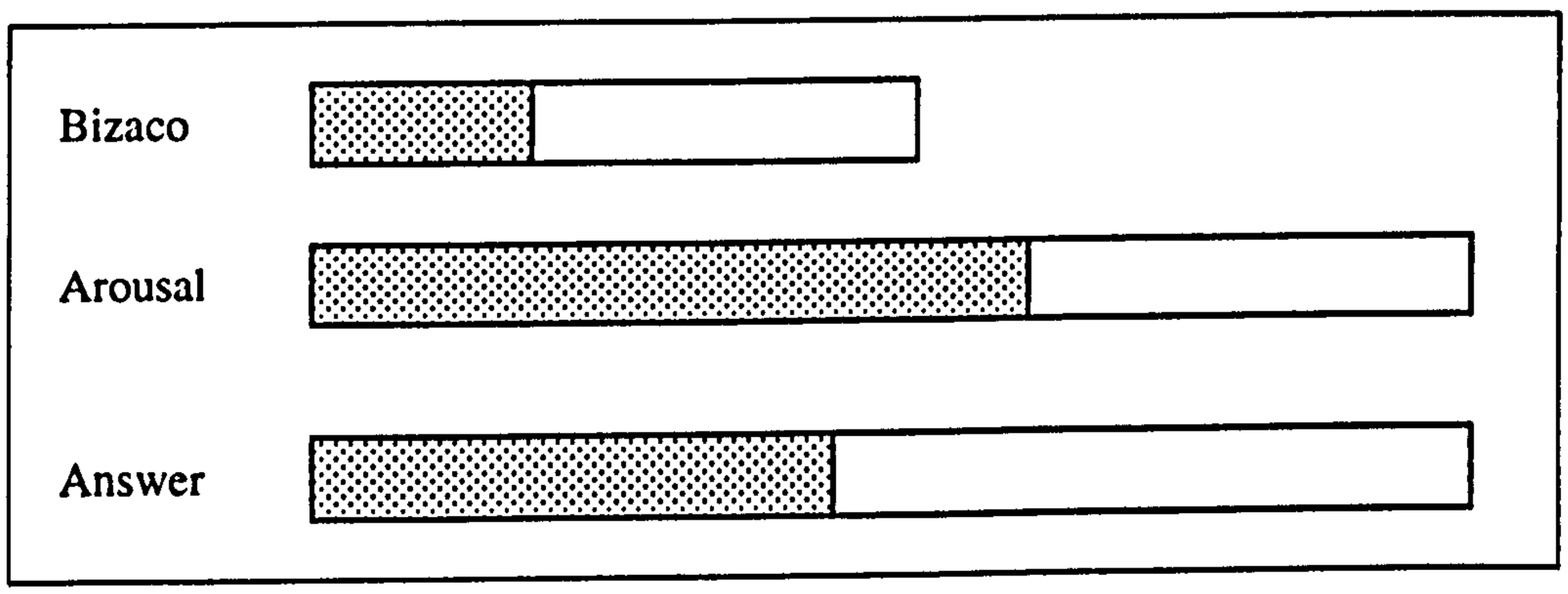


Figure 4.3 Scaled-down version of the bars that were used to represent input and output magnitudes. The upper bar represents the input to the system, the middle bar is participant controlled output, and target output values are presented in the lower bar. During testing, the target bar is absent.

4.2.1 Experiment 1

Linear relationships seem to hold a special place in our functional repertoire. For example, Sawyer (1991) has demonstrated that when participants are given a neutral cover story, they begin experiments assuming that the input and output dimensions conform to a linear function. Furthermore, several authors have demonstrated that linear functions are learnt more quickly than non-linear functions (e.g. Brehmer *et al.*, 1974; Byun, 1995; Naylor & Clark, 1968), and that extrapolation is approximately linear (Delosh *et al.*, 1997). Given these findings, one way in which we might learn functional relationships is to assume that the whole curve is made up of linear splines. The simplest version of this algorithm is known as piece-wise linear interpolation, and assumes that known examples are joined up with straight lines (an example is shown in the first panel of Figure 2.1, Chapter 2). Although EXAM is capable of producing linear interpolation in some situations, such as when all the training points fall in a straight line, it is not always the case that the model includes this interpolation in its range of response patterns.

Figure 4.4 illustrates a set of stimuli for which EXAM is unable to interpolate linearly. The circles mark the training points and the crosses mark the responses participants would make if they were responding with linear interpolation. The lines represent EXAM's predictions at different values of the discriminability parameter, λ. When λ is low, EXAM's responses are almost a straight line: there is very little local variation in the curve because there is always a large effect of

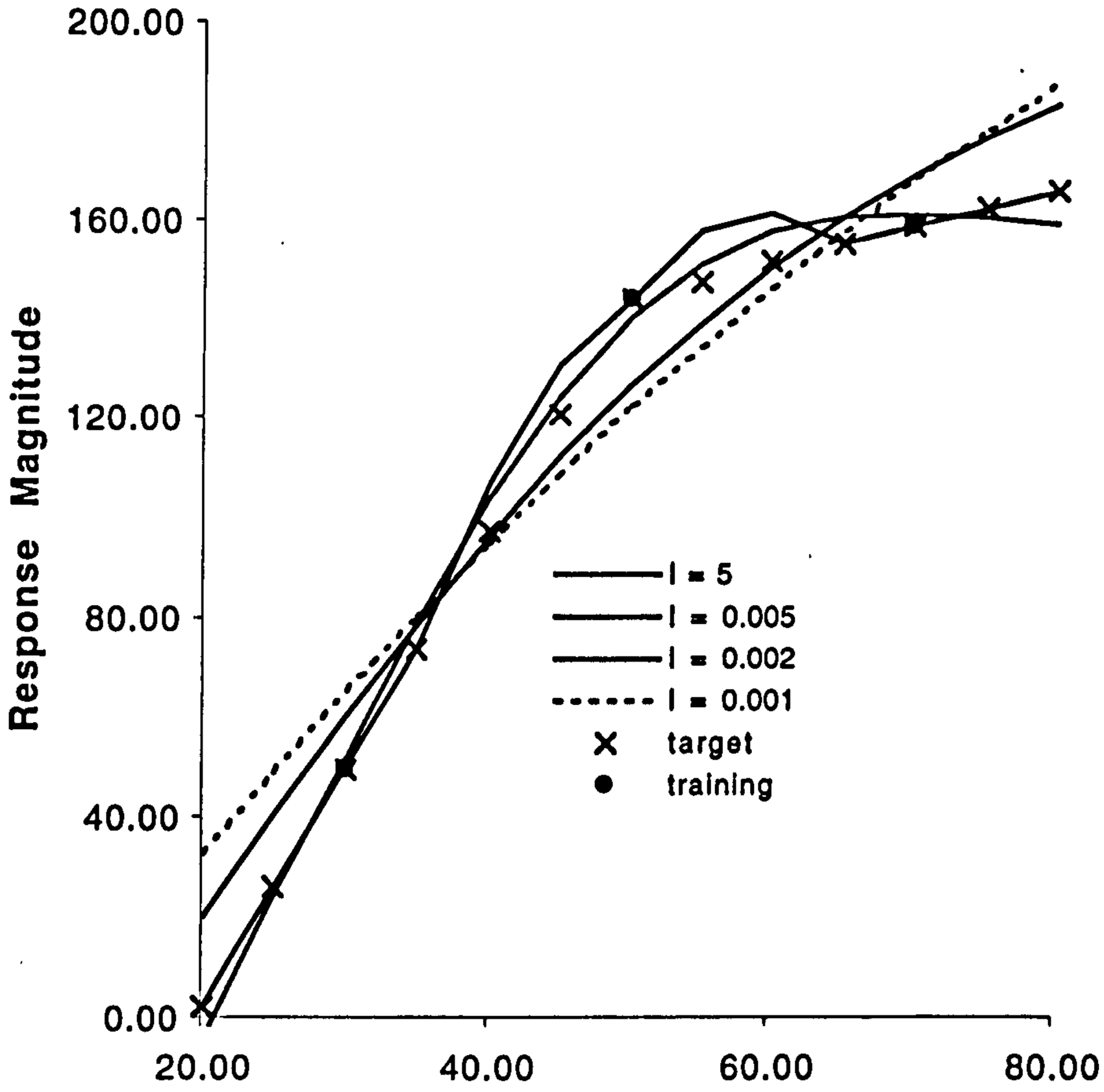


Figure 4.4 EXAM's predictions for the stimuli used in Experiment 1, as a function of λ ('l' in the legend).

all three training stimuli. As λ increases, the training points are reproduced more accurately but there is insufficient influence of the pattern as a whole. For example, consider the most discriminating λ value, $\lambda = 5$. At the furthest left training point, the curve initially follows the linear spline path, but then diverges as it approaches the central training point. The reason that it begins at this angle is that, here, the gradient of the anchor line is determined by drawing a line between the anchor and central training point (see Equation 9, where X_{i-1} is

replaced with X_i) and only this left-most training point has any chance of being selected as the anchor. However, as X increases, the central training point becomes entirely dominant. Because its anchor gradient is determined by the leftmost *and* the rightmost training points, responses move away from piece-wise interpolation. Thus, the line appears to be a series of local effects, rather than a continuous curve.

Past research on function learning have all used stimuli which EXAM can interpolate appropriately (as Delosh *et al.*, 1997, demonstrated) because of the high density of training points within the stimuli range. The stimuli magnitudes shown in Figure 4.4 are sufficiently sparse that EXAM is forced is interpolate over a much larger range, thus making its interpolation behaviour much more exaggerated. In Experiment 1, these stimuli are presented to participants. If participants are using a piece-wise linear interpolation, then it is predicted that a linear interpolation system will fit better than EXAM.

A second reason for carrying out this experiment is simply to explore the methodology developed by Byun (1995), Busemeyer *et al.* (1997) and Delosh *et al.* (1997). In their experiments, input and output magnitudes were presented as horizontal bars on a computer screen with length of bar corresponding to magnitude. At extremes of the extrapolation range, Delosh *et al.*'s participants appeared to show consistent departures away from the function to be learnt. They suggested that EXAM could explain the underestimation because, under the Hebbian learning algorithm, the training stimuli at the edges of the training domain were learnt less well than those in the centre. These stimuli then bring

down the slope of the extrapolation line, leading to the underestimation. An alternative explanation is that participants exhibited some bias when responding at the ends of the stimuli scale, that is, when they got near the ends of the bars shown in Figure 4.3. If this were true, it would imply more of a top-down, paradigm specific effect, rather than something that a generic model of function learning would be required to capture.

To examine this possibility, two between participant conditions were run. In the first condition, participants learnt the stimuli in a triangle configuration as described above. In the second condition, the input-output pairs were arranged in a straight line (the Linear condition). Both sets of stimuli are displayed in Figure 4.5, together with testing magnitudes and labels describing a High Extrapolation and a Low Extrapolation region. Notice that in the High Extrapolation region, the expected responses in the Linear condition are closer to the boundaries of the response bar (maximum 200) than the Triangle condition, whereas in the Low Extrapolation region, the expected responses in the Triangle condition are closer to the minimum of the response bar (zero). If participants are biased against the extremes of the bar, then a crossover effect is expected in terms of the deviation away from the expected extrapolation: the deviation will be greater the closer the linear extrapolation is to the extremes.

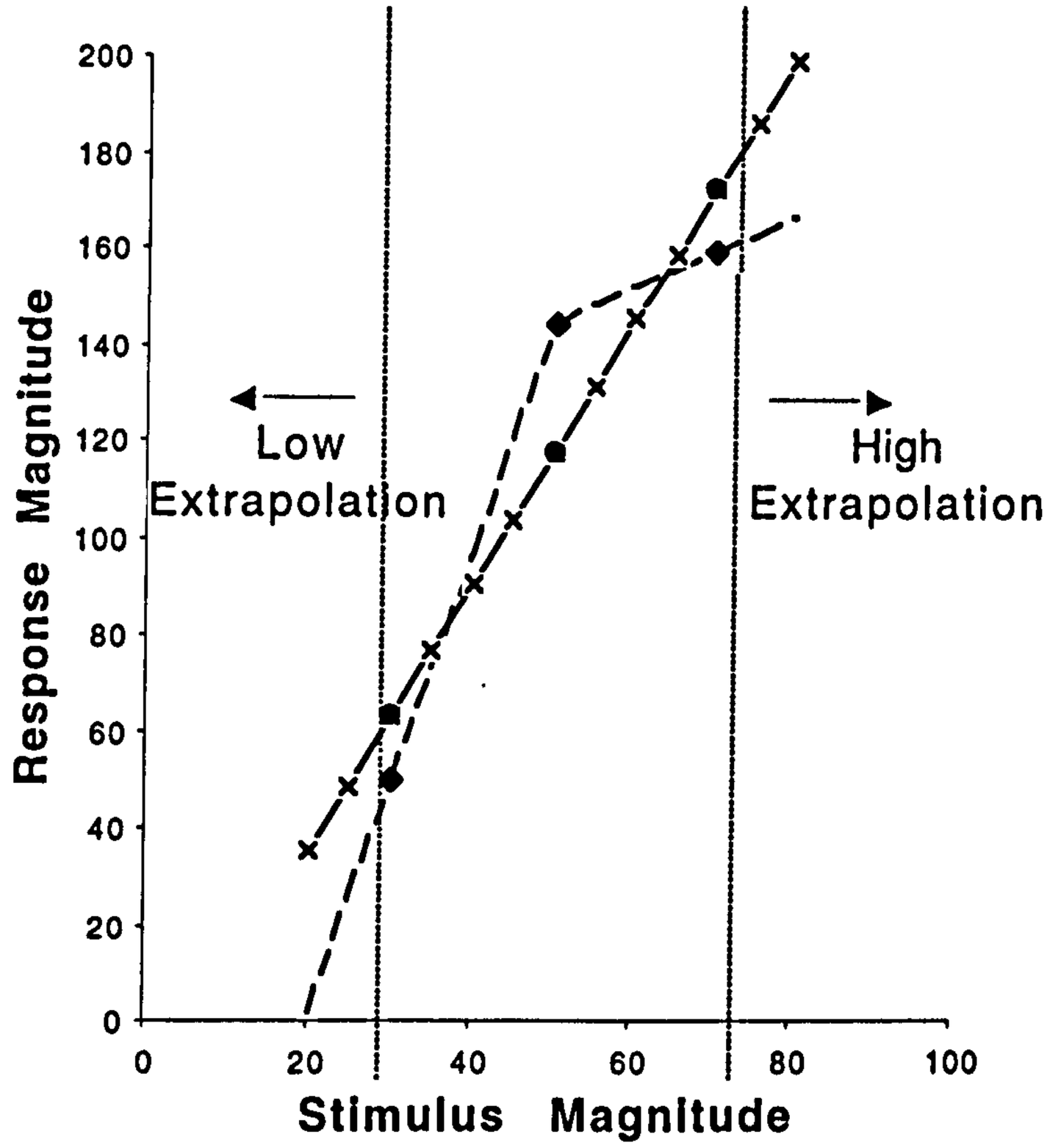


Figure 4.5 Stimuli for Experiment 1. Solid shapes correspond to training data, crosses to testing magnitudes (test items are only shown on one line for clarity). The straight line refers to the Linear condition, while the angled line refers to the Triangle condition. Regions are separated by dashed, vertical lines.

Method

Participants and apparatus

Twenty-eight Warwick undergraduates participated in the study for course credit. Fifteen participants were randomly allocated to each condition. Stimuli were presented on a colour, 35cm Macintosh monitor, with participants sitting about 60cm away from the screen.

Design and Stimuli

Participants went through training blocks and testing blocks. In the training blocks, participants were given an input magnitude and asked to respond with the appropriate output magnitude. After they had made their decision, they were provided with feedback in the form of the correct output level for 1.5 seconds. They then proceeded onto the next example. The three input magnitudes presented were 30, 50 and 70. In the Linear condition the target values were 63, 118, and 173 respectively. In the Triangle condition, the values were 50, 145 and 160. Each block consisted of a single presentation of each of the 3 examples. After a training block, participants moved onto a test block where they were not provided with feedback. Here, they were tested on input magnitudes varying from 20 to 80, in increments of 5, a total of 13 magnitudes. In both training and testing blocks, stimuli were presented in a random order. There were 10 training blocks, each one being followed by a testing block.

Following the paradigm developed by Delosh *et al.* (1997), stimuli were presented and recorded as three, red and blue horizontal bars placed one above the other (as shown in Figure 4.3). The first bar showed the input to the function, the second the user-defined output, and the third showed feedback (the correct output). The magnitude of the function values were given by the proportion of the bar which was red. For example, to indicate a feedback output of 150 (out of 200), the lowest bar was three-quarters red, and one quarter blue. In addition, bars were labelled to correspond to the cover story. Like Delosh *et al.*, the input bar ranged from 0 to 100, whereas the other bars varied from 0 to 200 and were correspondingly twice as long.

Procedure

Participants were first presented with the following instructions: "In this experiment, we'd like you to learn the relationship between the amount of a drug (called Bizacol) and the level of arousal caused by taking the drug. You will be presented with examples of the quantity of Bizacol taken, and the corresponding level of arousal. Your task is to learn these examples by a process of trial and error and the feedback provided by us". They were then instructed to use the arrow keys to alter the response bar and to hit SPACE when they had made their decision. The timing was entirely at the participant's discretion, but they were told that although there was no time limit, they should not spend more than about 10 s on any one trial.

Results and Discussion

Inferential statistics

Inferential statistics were carried out on the absolute deviation of participants' responses from the target values. For the training data, the target values were the feedback magnitudes provided. For the testing data, these were the piece-wise linear generalisation values shown by the lines in Figure 4.5. The lower the error score therefore, the closer the participants' responses to these response curves.

The mean absolute error scores (MAE's) for the two different conditions are shown in Figure 4.6. These are the scores for the testing phase (which included input magnitudes presented in training) only. Participants seem to reach a plateau at about the fifth block and both conditions appear to have learnt the data to the same extent. This intuition was confirmed by an ANOVA based on a cubic transformation of the means to homogenise variances. This revealed a main effect of Block, $F(9,252) = 11.34$, $MSE = 0.04$, Huyn-Feldt Epsilon = 0.68, $p < 0.0005$, but no interaction, $F(9,252) = 0.57$, nor main effect of Stimulus pattern, $F(1,28) = 2.98$, $MSE = 0.47$, $p = 0.095$. However, performance in the Triangle condition was worse in 10 out of 10 blocks ($p < 0.01$ on a sign test), which does suggest that some effect of Stimuli. Indeed, a complete lack of an effect of Stimuli would be surprising, since Byun (1995) observed that functionally combined input-output pairings (that is, those conforming to a linear or quadratic pattern) are learnt more easily than randomly combined pairings. A

possible explanation for the weakness of the effect is that the 'random' condition (the Triangle pattern) was not sufficiently different to the 'functionally combined' condition (the Linear pattern) for the differences to manifest itself.

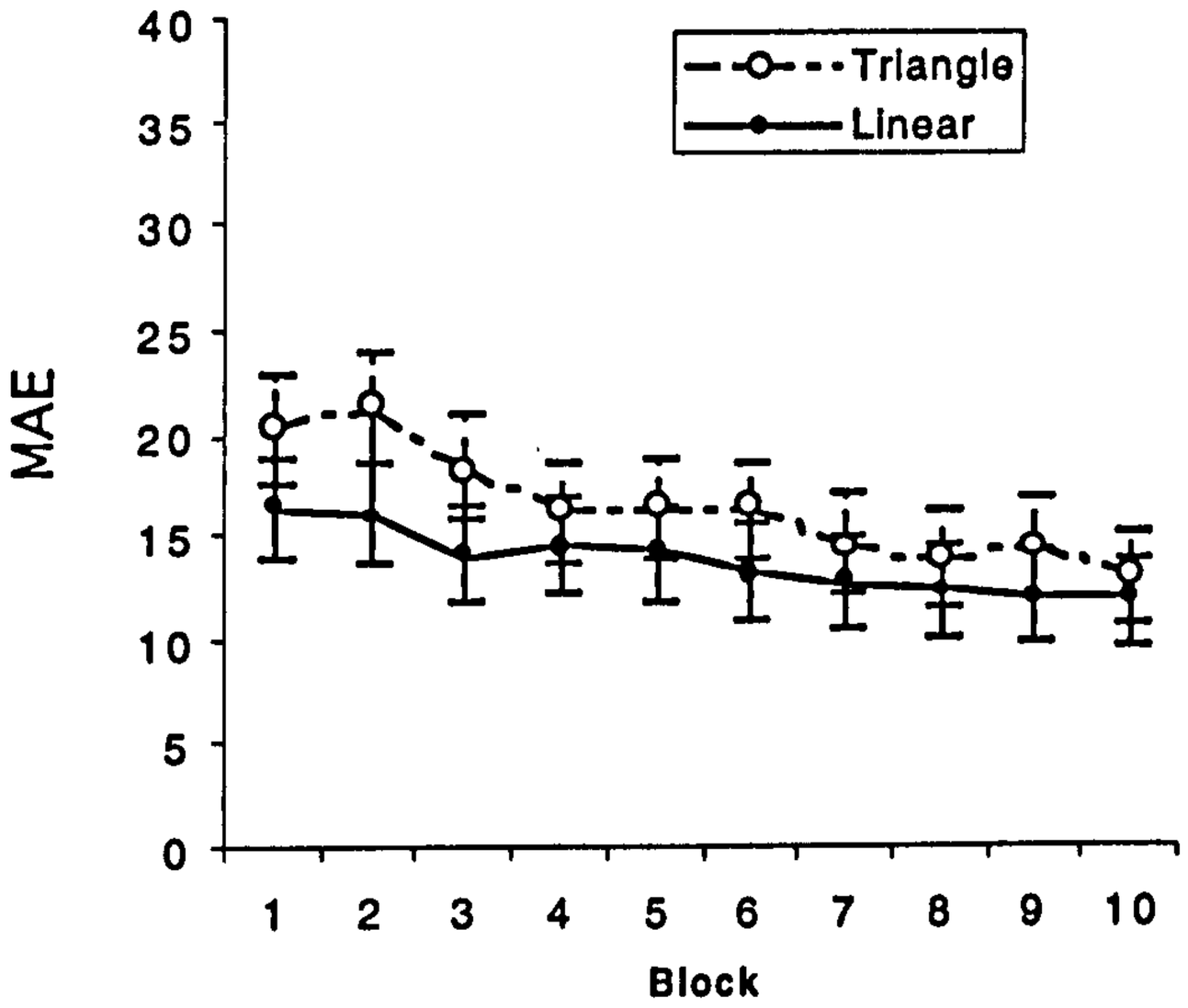


Figure 4.6 Mean absolute error as a function of Block and Stimulus Type. Error bars correspond to the standard error of each cell.

Figure 4.7 shows how participants' MAE's (over the last five blocks) vary as a function of x. In the Low extrapolation region, those in the Triangle group seem to deviate more from the linear extrapolation than those in the Linear group, whereas the situation is reversed for the High extrapolation region. This is the situation that was predicted if participants are biased against responding in the extremes of the response bars, illustrated by the evident crossover in Figure 4.8. An ANOVA confirmed the reliability of this conclusion, with a main effect of Range, $F(1,26) = 38.64$, $MSE = 23.78$, $p < 0.0005$, an interaction of Range and Stimulus Pattern, $F(1,26) = 111.03$, $p < 0.0005$, but no main effect of Stimulus Pattern, $F(1, 26) = 0.19$, $MSE = 50.24$, $p = 0.664$. Paired t-tests revealed that in

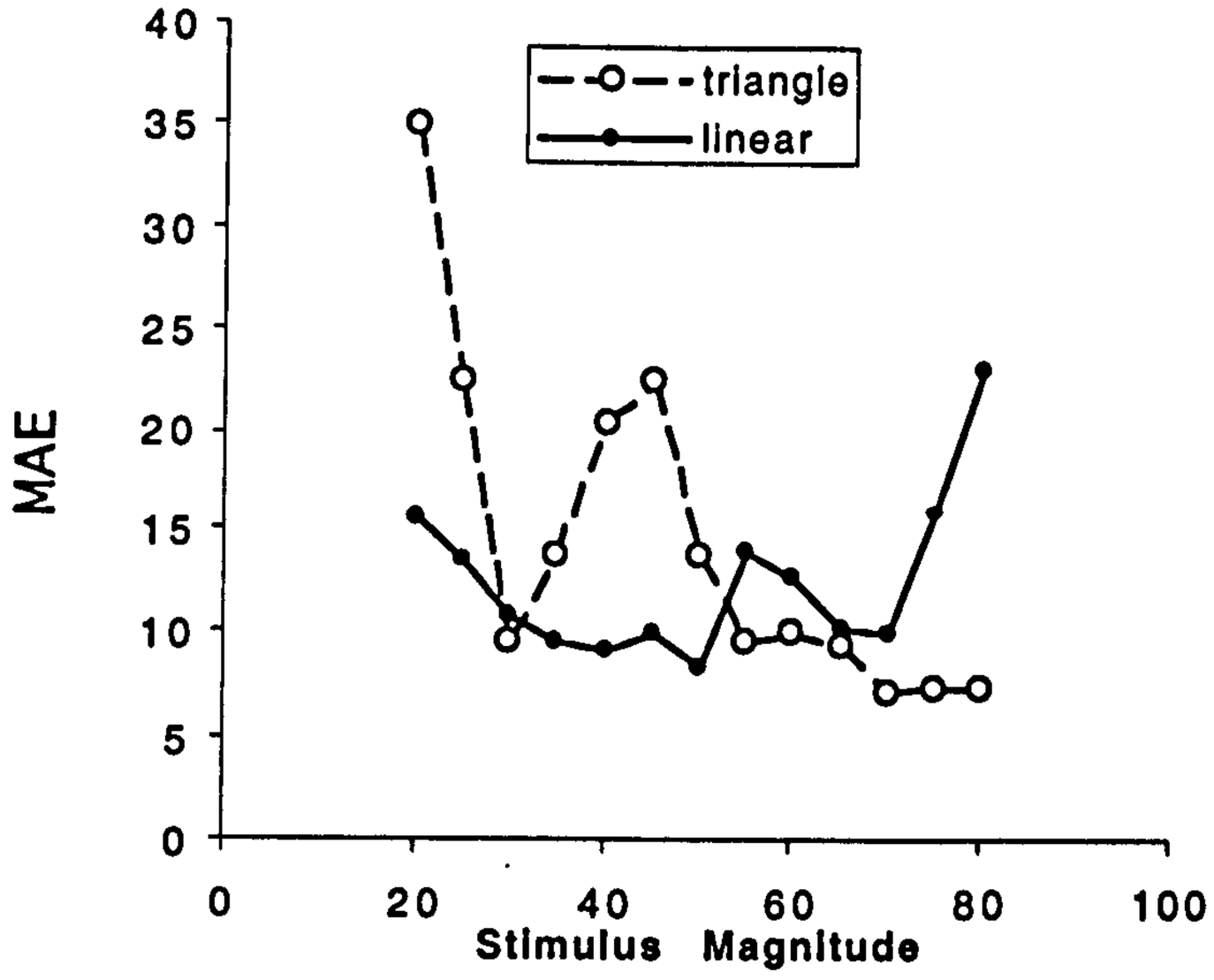


Figure 4.7 Mean absolute error as a function of Stimulus Type and Stimulus magnitude.

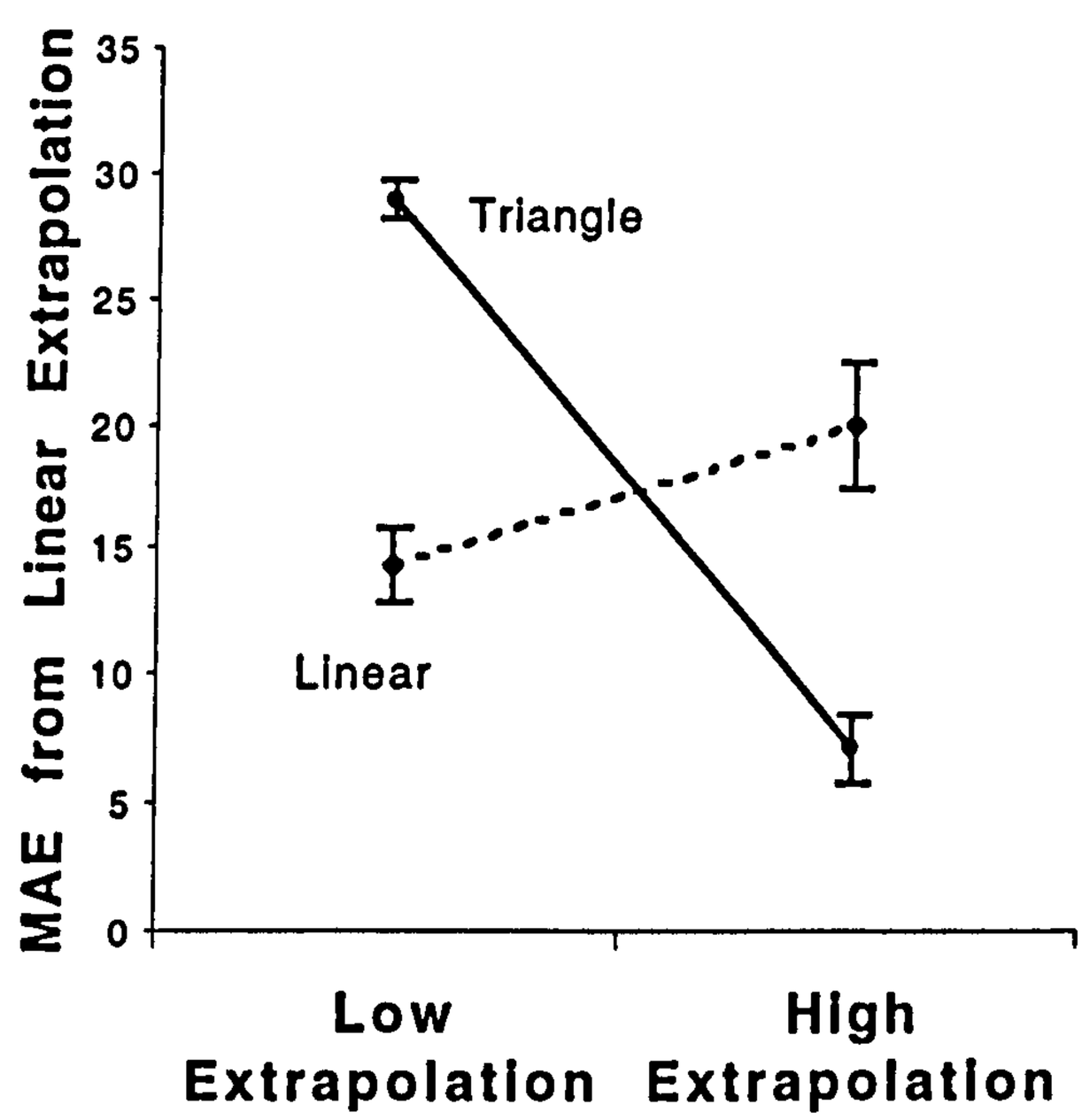


Figure 4.8 Mean deviation from linear extrapolation as a function of extrapolation region and stimulus set. Error bars refer to the standard error of the mean.

the Low extrapolation region, those in the Triangle condition differed by significantly more than in the Linear condition, $t(26) = 5.22$, $p < 0.0005$ and in the High region, those in the Linear condition differed more than in the Triangle condition, $t(26) = 7.74$, $p < 0.0005$. Further, the Interpolation region (which includes the training points) did not differ between the two Stimulus Patterns, $t(26) = 1.34$, $p = 0.19$, eliminating the possibility that differences in extrapolation were due to differences in the interpolation region.

Given that the extent of deviation depends on how close expected responses are to the extremes, it can be concluded that participants are biased against responding to the extremes of the measurement scale. This result may explain why Delosh et. al. (1997) found that participants underestimate the linear function in the extrapolation regions. Although they found that participants underestimate in both high extrapolation and low extrapolation regions, whereas participants in this study underestimate in the high regions but overestimate in the low regions, the form of these biases may well vary from sample to sample. Note that even if high-level biases are not the answer, Delosh *et al.*'s hypothesis cannot account for differences in observed bias between this study and their own.

Modelling

One of the goals of the experiment was to establish whether participants linearly interpolated or not. This was examined by fitting EXAM and a piece-wise linear model to the asymptotic responses of participants. The linear model consisted of

straight lines joining the three training points up, and a linear continuation into the extrapolation regions, as shown by the dashed line in Figure 4.5. Because EXAM and the linear model can make the same interpolation predictions in the Linear pattern condition, the models will only be fit to the Triangle condition. Fits were made to both group data and individual participants' responses.

The models were optimised on the average responses of participants over the last five blocks, that is, when learning was judged to be at asymptote. This is because the experiment is concerned with the generalisation properties of EXAM, not the learning mechanism. The measure of fit used was the r^2_{adj} [2] which takes into account EXAM's free parameter, λ. For both models, the r^2_{adj} was based on 13 (testing points) - 3 (training points) = 10 scores for the group responses, and 10 per participant for the individual fits.

Procedure

Optimising the linear interpolation model involved taking the participants' responses to the training input values, and forming a set of splines from these, rather than from the target values (the same is true for fitting EXAM). For example, for the group data, the average responses to training values 30, 50 and 70, were 54.9, 133.9, and 158.5. These values then formed the pivots of the final

---

[2] The r^2 adjusted is given by $r^2_{adjusted} = 1 - \left(\frac{\sum_{i=1}^{N} (y_{obs} - y_{pred})^2 / (N-k)}{\sum_{i=1}^{N} (y_{obs} - \bar{y})^2 / (N-1)} \right)$, where y_{obs} is the observed y response, y_{pred} is the response predicted by the model, $\bar{y}$ is the mean of the observed responses, N is the number of data points and k is the number of parameters of the model.

spline function. This was done because generalisation is assumed to be based on what participants perceive the training values to be, not on what the 'correct' output should be. Once the spline function had been estimated, the summed squared error (SSE) of the responses from this line was calculated and converted into an r^2_{adj} value.

EXAM has one free parameter, λ, which needs to be optimised. This was achieved by minimising the SSE between participants' responses and EXAM's predictions. However, EXAM's weights from the input to the output layer also have to be estimated (see Equation 3). This procedure involved choosing a λ value, then optimising the weights using gradient descent on the error between the participant's responses to the training input values and the model's predictions. For example, for the group data, EXAM was given the training points {30, 54.9}, {50, 133.9} and {70, 133.9}, and a λ value of, say, 0.01. The gradient descent algorithm was then used to adjust the weights so that EXAM produced the training values. Once this had been done, EXAM was tested on the input values from the test phase (not including the values presented in training) and an SSE was calculated between participants' testing responses and EXAM's predictions. λ was optimised on this SSE score using the 'fmin' function in the MATLAB 5.2 Toolbox. The gradient descent procedure used a sufficiently low learning rate, 0.05, that EXAM could reproduce the training values within an average of 7 units (despite the fact that most of the best-fit λ values prevented the model from reproducing the training data accurately, as in the $\lambda = 0.001$ line in

Figure 4.4). Furthermore, the error from the training value was not included when calculating the r^2_{adj} scores.

Results and Discussion

For the group data, the r^2_{adj} for the linear interpolation model was found to be 0.97, while for EXAM it was 0.96 with a best-fitting λ of 0.0011. Responses of the models, together with participants' average responses are shown in Figure 4.9. Note that EXAM's responses are essentially a smooth curve over the entire range, due to the very low discriminability value, which doesn't capture the qualitative pattern of the responses.

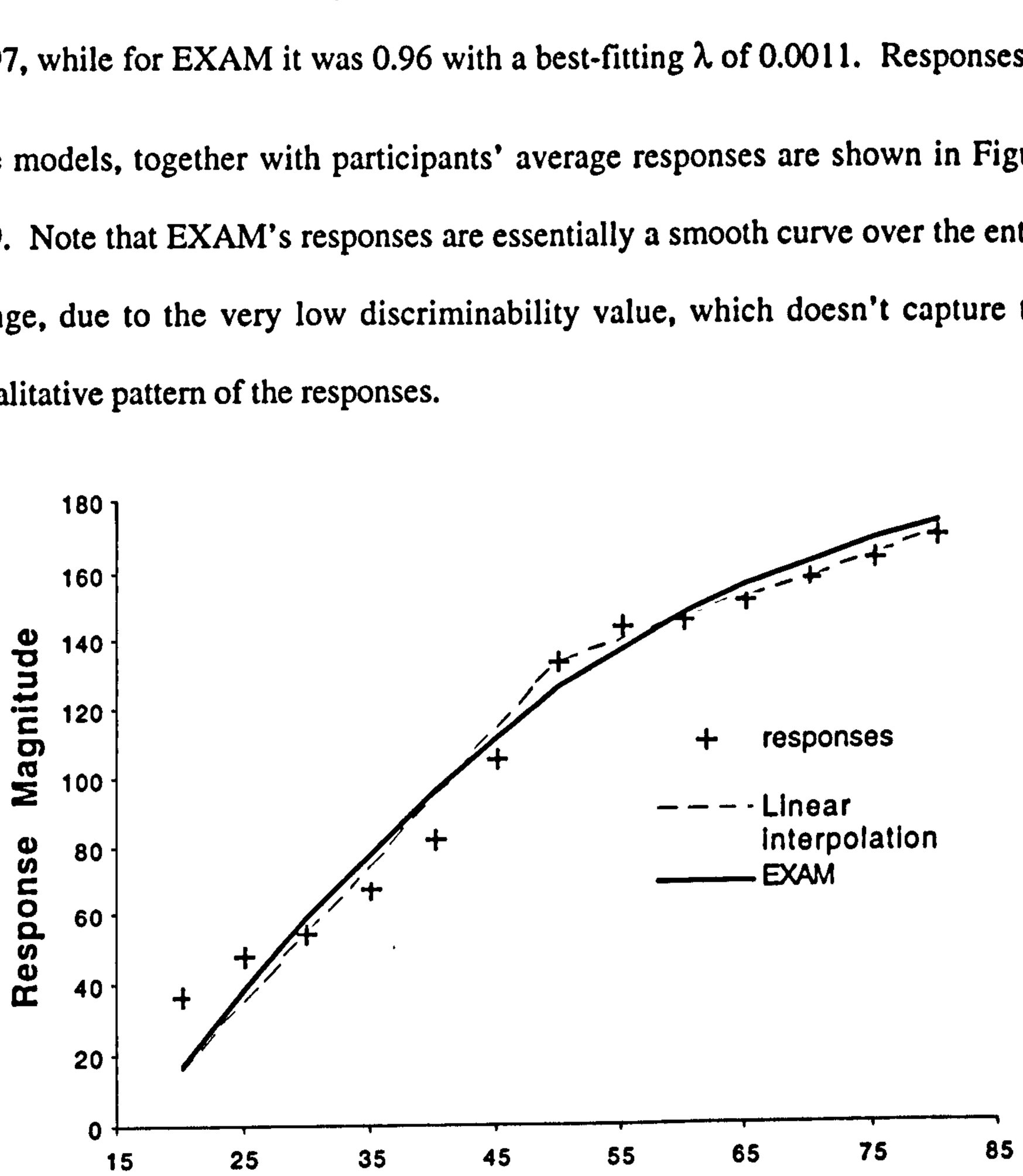


Figure 4.9 Response magnitudes for EXAM, the linear interpolation model, and participants' mean responses at asymptote (Triangle condition).

Table 4.1 displays the r^2_{adj} values for the individual data, together with the best-fitting λ values for EXAM. These fits illustrate similar patterns to the group fits, with approximately equal r^2_{adj} values for the two models and λ's of similar magnitudes.

participant	r^2_{adj} Linear Interpolation	r^2_{adj} EXAM	λ (× 10^{-2})
1	0.93	0.90	0.12
2	0.82	0.80	0.16
3	0.92	0.92	0.18
4	0.87	0.86	0.12
5	0.76	0.71	0.16
6	0.95	0.96	0.17
7	0.69	0.80	0.10
8	0.92	0.86	0.13
9	0.97	0.97	0.16
10	0.97	0.96	0.12
11	0.93	0.92	0.09
12	0.93	0.91	0.09
13	0.91	0.91	0.11
14	0.94	0.94	0.11

Table 4.1 Fits of the linear interpolation model and EXAM to individual participants' responses over the last five blocks of testing.

The linear interpolation model seems to capture the data as well as EXAM (if not better), although its performance on other data sets remains to be seen. One disadvantage with a piece-wise linear model is its susceptibility to noise in the data set. If each data point is joined together by a spline, then the result would be a very jagged curve and poor generalisation, rather like the first panel in Figure 2.1. EXAM avoids overfitting by altering the smoothing parameter or the learning rate, neither or which are present in the linear model. A solution to this would be to move away from a piece-wise function, to one with a restricted number of linear functions joined together (the precise number being a free

parameter). The problems with noise could then be absorbed by the number of splines used to fit the function: a higher perception of noise in the training data would lead to fewer splines being fit. This method would have the advantage of fitting the data in this experiment and using a more elegant extrapolation system. Further work could formalise this suggestion and test the two versions on other data.

To summarise, this experiment has established two important points about function learning. First, participants seem to shy away from the extremes of the response bars. This means that, regardless of whether Delosh *et al.*'s (1997) participants displayed a similar bias, care should be taken when examining extrapolation responses within this paradigm. Secondly, evidence was provided that a linear interpolation model provided just as good a fit to the data as EXAM. Furthermore, the linear model seemed to give a better qualitative account of participants' responses. The issue of linear interpolation will be returned to in the General Modelling section of this Chapter.

4.2.2 Experiment 2

Consider again Figure 4.2. The extrapolation response from EXAM is a weighted sum of the straight lines leading from the data points. These weights are the probabilities of selecting each training point to be the anchor. This means that, if the probabilities remain constant as the distance away from the training points increases, then extrapolation is linear. The probabilities are determined by a Gaussian function of the distance between the testing stimulus and the training points (Equation 8). As the Gaussian function flattens out, therefore, the probabilty of choosing any given training point as the anchor becomes constant as the distance increases.

Consequently, the model can never predict any curve which has a cyclic nature, because the gradient of a cyclic curve is not constant as x increases. This is not to say that EXAM can only predict linear responses - at short distances, with appropriate parameter settings, EXAM can predict even nonmonotonic extrapolation. However, this is a very unlikely situation – EXAM was constructed to show linear extrapolation at asymptotic learning. Figure 4.10 demonstrates EXAM's behaviour with a variety of different scaling parameter values, after having been trained on data from a cosine curve. As a guide, Delosh *et al.* used a λ value of about 2 to fit their participants' data. This figure demonstrates that extrapolation only departs by a fraction from linearity regardless of the λ parameter, assuming the data have been learnt to a reasonable degree.

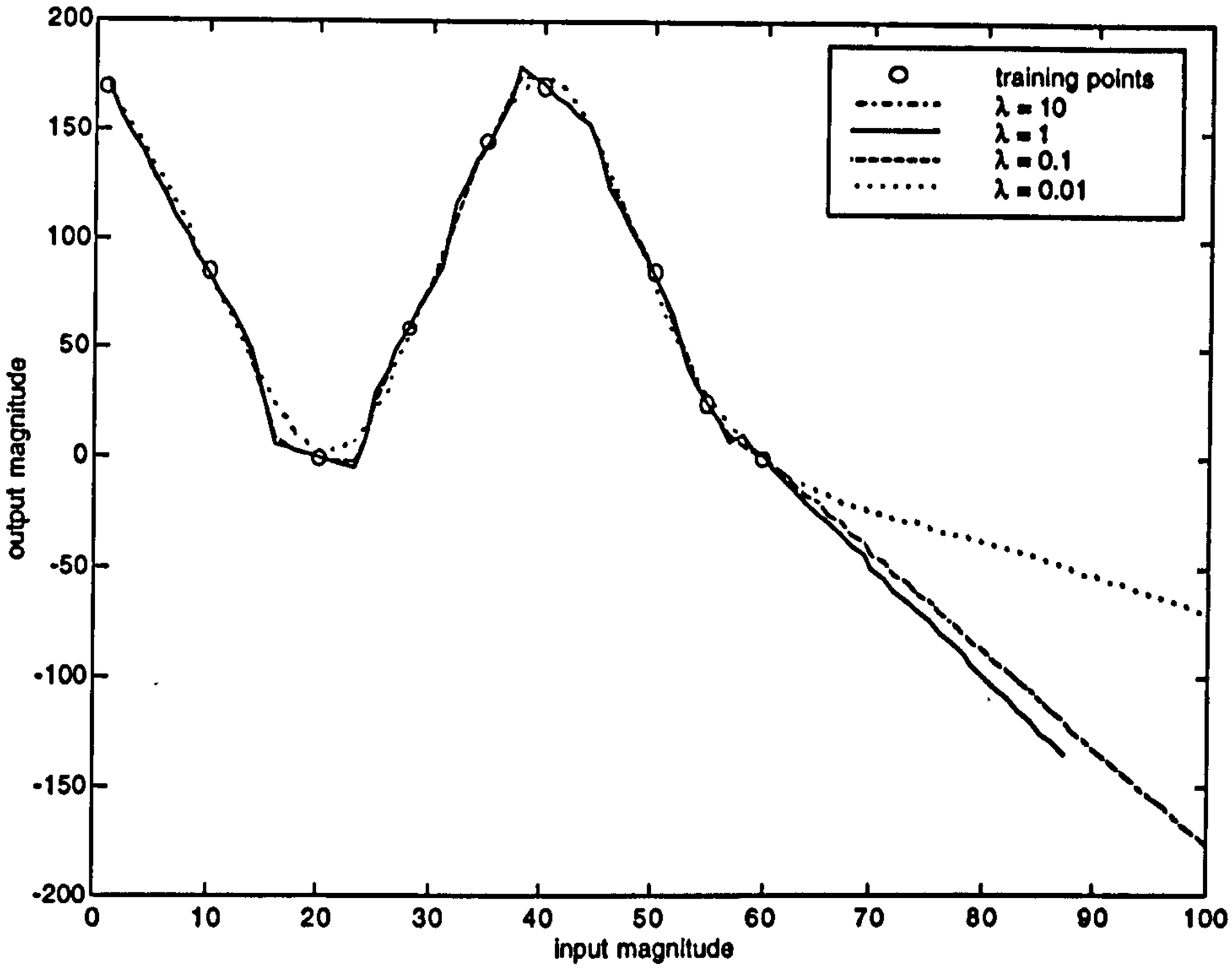


Figure 4.10 EXAM's responses to the training data as function of λ

However, people might well be expected to extrapolate nonlinearly in the right situation. For instance, there are few people who would not predict that the height of the sun will increase tomorrow morning and decrease tomorrow evening, or that a pendulum will continue to swing back and forth, or that, next year, the number of people taking holidays in the summer will be higher than the number in the winter. These examples are all forms of cyclic curves, which suggests that they might provide a suitable nonmonotonic function with which test EXAM. Indeed, Estes (1984) demonstrated that participants continued to expect probabilities of success to vary cyclically long after feedback suggested

otherwise. His result provides evidence against EXAM if we extend EXAM's predictions outside the paradigm in which it was developed. However, given the effects different instructions and stimuli can have on participants' responses, it is perhaps wise to transfer Estes's (1984) experiment to the paradigm used by Delosh *et al.* (1997)[3].

Participants were presented with a series of examples which describe inputs and outputs from a cosine curve. Following training, they were tested on input values beyond the range that they had previously encountered (extrapolation). It was predicted that participants would continue the cosine curve, rather than extrapolating linearly.

Participants were first tested on stimuli that they had had no training on, in order to assess their understanding of the cover story. After this initial testing session, they were presented with 12 training-testing blocks. During training, they were presented with the same 9 examples of the curve on each block, followed by testing on 9 interpolation points and 6 extrapolation points (see Figure 4.11 below). In this experiment, there were two departures from the methodology used by DeLosh *et al.* (1997). These changes were designed to make the training task easier for the participants, because it was thought that learning a cyclic curve was more difficult than the curves that Delosh *et al.*'s participants learned. First, the points were presented sequentially in ascending order of the input, so that, for example, $x = 2$ is presented before $x = 7$, before $x = 14$. Byun (1995) has

[3] Although Byun (1995) has shown that participants are capable of learning examples that conform to a cyclic curve during training, no attempt was made to test their extrapolation. Therefore, her results do not pose a problem for EXAM.

showed that participants trained on sequential orders learn more quickly than those trained on the random order used by Delosh *et al.* Further, Estes's (1984) experiment was a form of sequential learning. Secondly, participants were given a cover story which suggested a cyclic curve, as opposed to the neutral cover story presented by Delosh *et al.* Again, Byun (1995) demonstrated facilitation with congruent cover stories. These differences are addressed in the discussion of this experiment and examined further in Experiment 2.

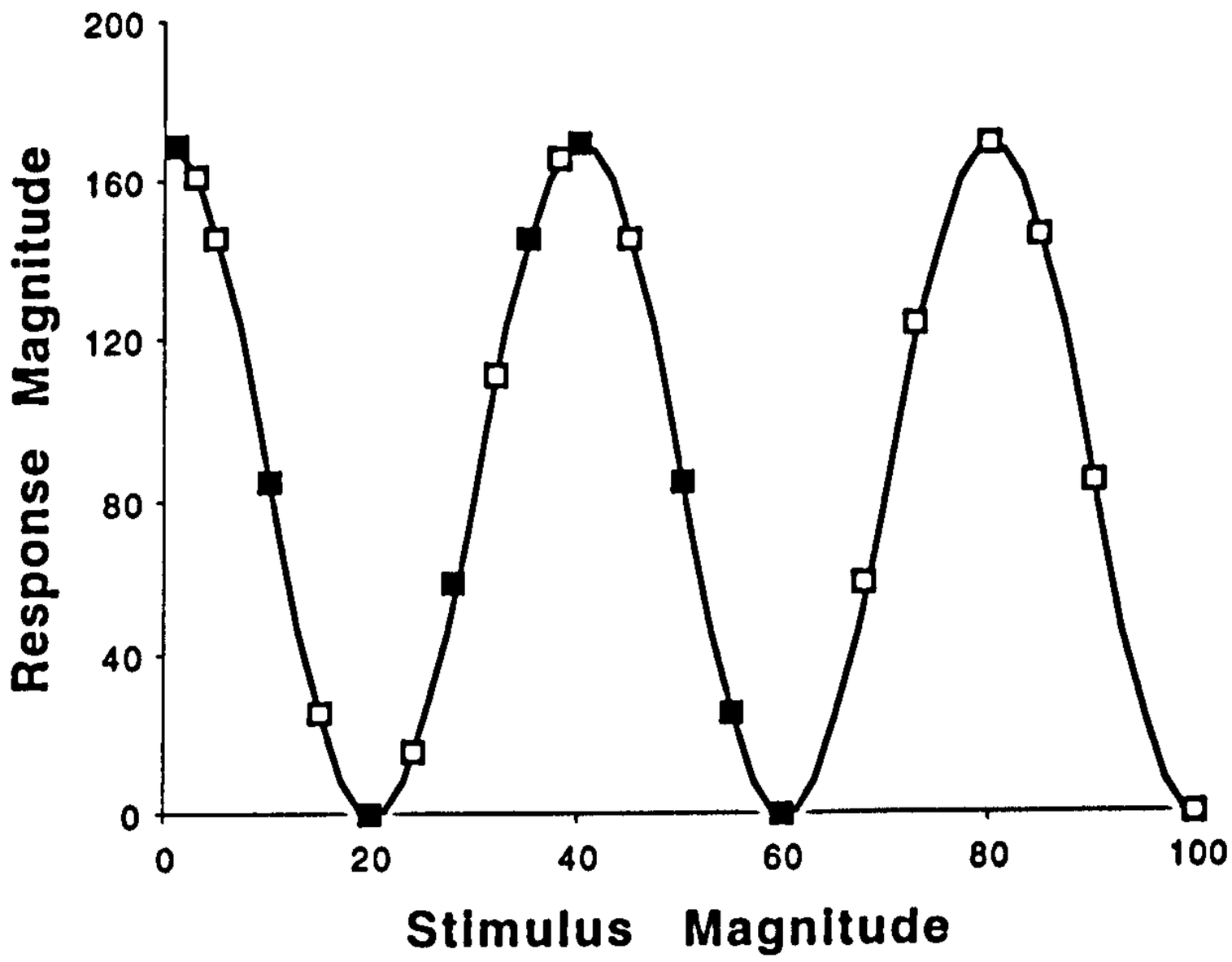


Figure 4.11 Training and testing values for Experiment 2. Filled squares indicate training values; empty squares indicate testing values.

Method

Participants and apparatus

Twelve University of Warwick students were paid £5 for their participation in the experiment, which took less than an hour to complete.

Design and stimuli

In each block of the training phase, participants were given input values and asked to predict the output. Feedback indicating the correct output was provided after each trial. There were 9 trials in each block. In the test phase, they were presented with 15 input values, but no feedback was provided. Of these, 9 were interpolation points, and 6 were extrapolation. The input range for both training and testing was 0 to 100, and the output range 0 to 200 (as shown in Figure 4.3). Figure 4.11 illustrates the training stimuli. The function which generated the input and output pairs was $y = 85 + 85\cos(x\pi/20)$.

There were 12 complete blocks, each block consisting of a training phase, followed by a testing phase. In addition, there was one testing block before any feedback had been given.

All the points were presented in sequential order within each phase (training or testing) and the same set of points was presented in each block.

Procedure

Participants first read the following instructions: "In this experiment, we'd like you to learn to predict the height to which a ball will bounce after a certain time. You can imagine that there is a person who bounces the ball continuously over a

time period, and you have to predict the height of the ball after a certain length of time. The ball will start off in their hand, and be bounced up and down for a number of times.

You will be presented with a set of examples, each example consisting of a height that the ball bounces at, and the length of time since the ball started bouncing. Your task is to learn the relationship by a process of trial and error and the feedback provided by us".

All other aspects of the of the methodology are identical to Experiment 1.

Results and Discussion

To assess whether participants had improved over training, the training data were analysed using a repeated measures ANOVA on the absolute deviation of the responses from the target magnitudes, with block as the only factor. This yielded a significant effect F (11,121) = 27.53, Huynh-Feldt Epsilon = 0.46, MSE = 95.68, $p < 0.0005$. The mean deviations are plotted as a function of block number and participant in Figure 4.12. The bottom curve shows the average error from all participants except numbers 8, and 12, who are plotted above on separate curves. The majority of participants learned the data well, showing the familiar exponential drop in training error, but Participants 8 and 12 learnt comparatively little. Much of the analysis is done on an individual basis, but where there is examination of the data in general, participants 8 and 12 are included.

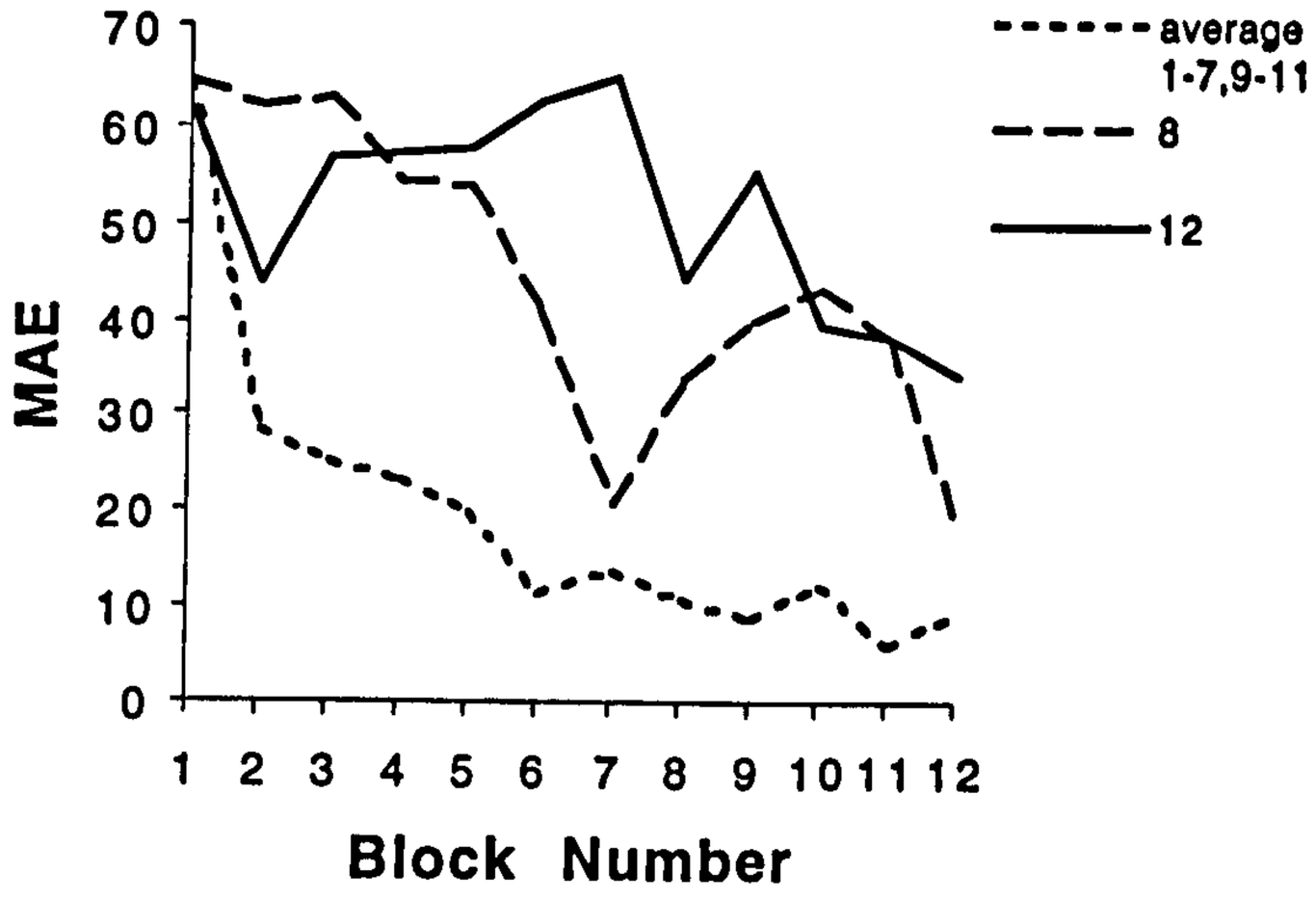


Figure 4.12 Mean Absolute Error as a function of Block and Participant for Experiment 2.

The extrapolation data are slightly more complicated to analyse. Informal examination of the graphs of stimulus magnitude by responses reveal that most participants extrapolated in a cyclic way, continuing with the pattern they demonstrate in training. The first four testing blocks and the last four testing blocks of Participant 1 are shown in Figure 4.13, to give a general idea of the responses (the middle blocks were very similar to the last four). The dotted line is the target function, and the crosses represent responses by the participant. The vertical dashed line illustrates the boundary between interpolation and extrapolation. Participant 1 can be considered representative of about two thirds of the participants, while the others will be examined in more detail later on. As can be seen from the graph, the participant initially responds in a linear fashion, but as more blocks are experienced the pattern in both interpolation and extrapolation becomes cyclic. Noticeably however, the cycles are never quite in phase, which means that average deviations from targets could be misleadingly high. Thus, participants may well be moving towards a cyclic function, although

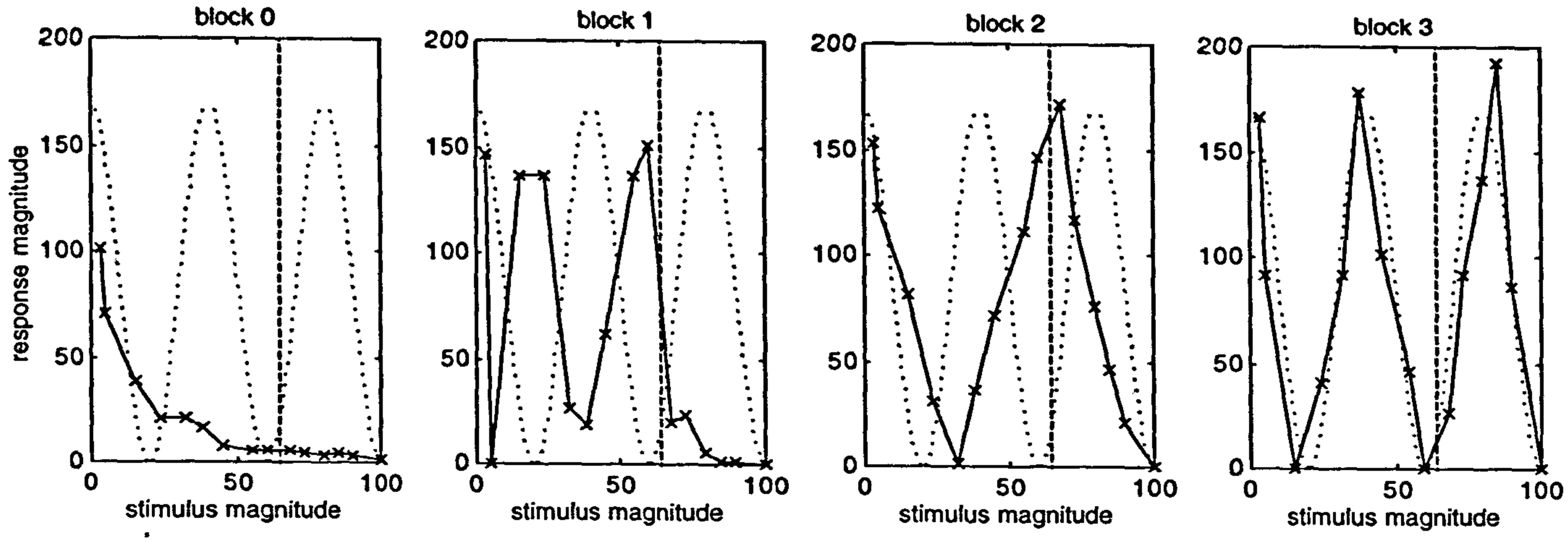


Figure 4.13 Participant 1's responses to the training and test data. Crosses to the left of the vertical dashed line are training points, while those to the right are testing points.

the difference between the deviations in early blocks may not be lower than in later blocks. This problem implies that performing an ANOVA on absolute deviations from the target function, as we did for the training data, would be inappropriate. One solution is to fit a function to each block of each participant, and find the deviations from this line of best fit. This is the analysis that was carried out below.

Before examining this issue however, it is useful to look at the four participants who did not perform in the standard way. Figures 4.14 - 4.17 shows testing blocks from Participants 4, 7, 8 and 12 respectively. These will be dealt with in turn. Participant 4 has learnt the training data to a reasonable degree, and if we look purely at the extrapolation section most of the blocks indicate a truncated V shape. However, looking at the whole range, the W shape apparent in his responses probably indicates that he is underestimating the frequency, and thus responding monotonically in *his* conception of the extrapolation range (contrary to our hypothesis). Of course, if he were tested on even more extreme x-values, he might well have shown a decreasing function after the $x = 100$ point. Some evidence for this is provided by the fact that he approaches (see Blocks 3-8), but never exceeds, the amplitude in the extrapolation region, and might decrease from this given the opportunity. Participant 7 (Figure 4.15) again seems to have learnt the training points, but in contrast to Participant 4 appears to *overestimate* the frequency of oscillations. This leads her to have a slight, but consistent, upward tail towards the higher values of the extrapolation section, in addition to the inverted V we would expect. Participant 8 has a large training error (see Figure 4.16), but by the end of training seems to show evidence of having learnt

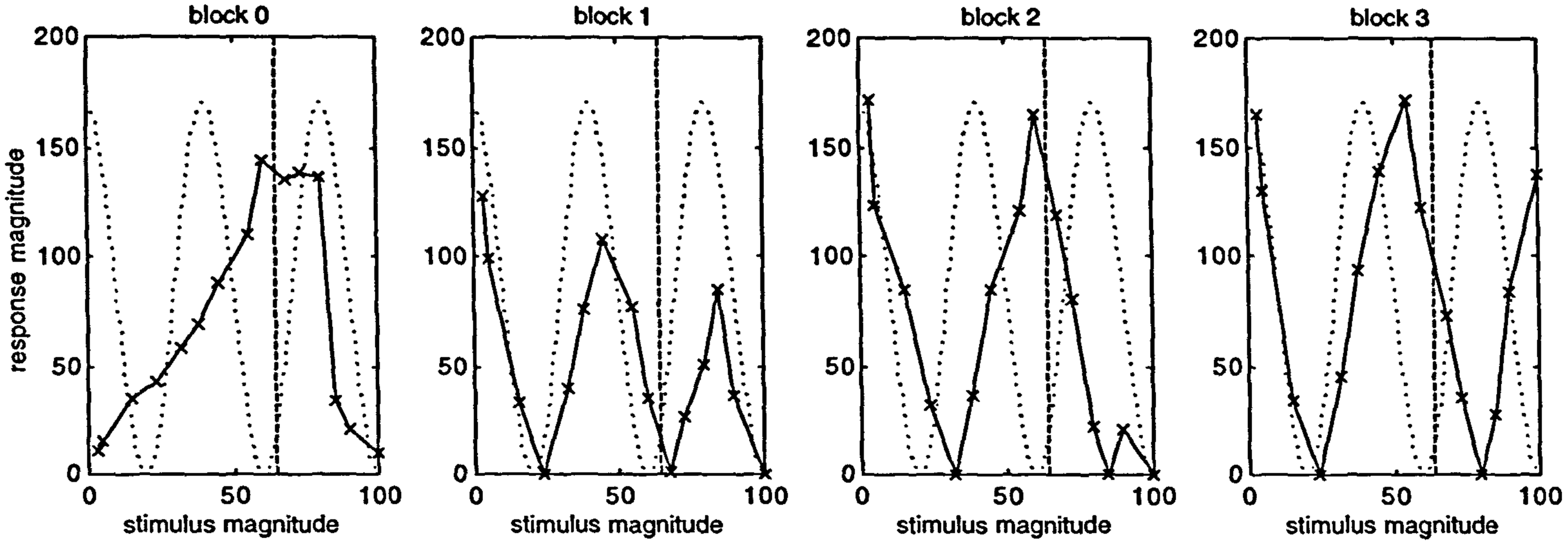


Figure 4.14 Participant 4's responses to the training and test data. Crosses to the left of the vertical dashed line are training points, while those to the right are testing points.

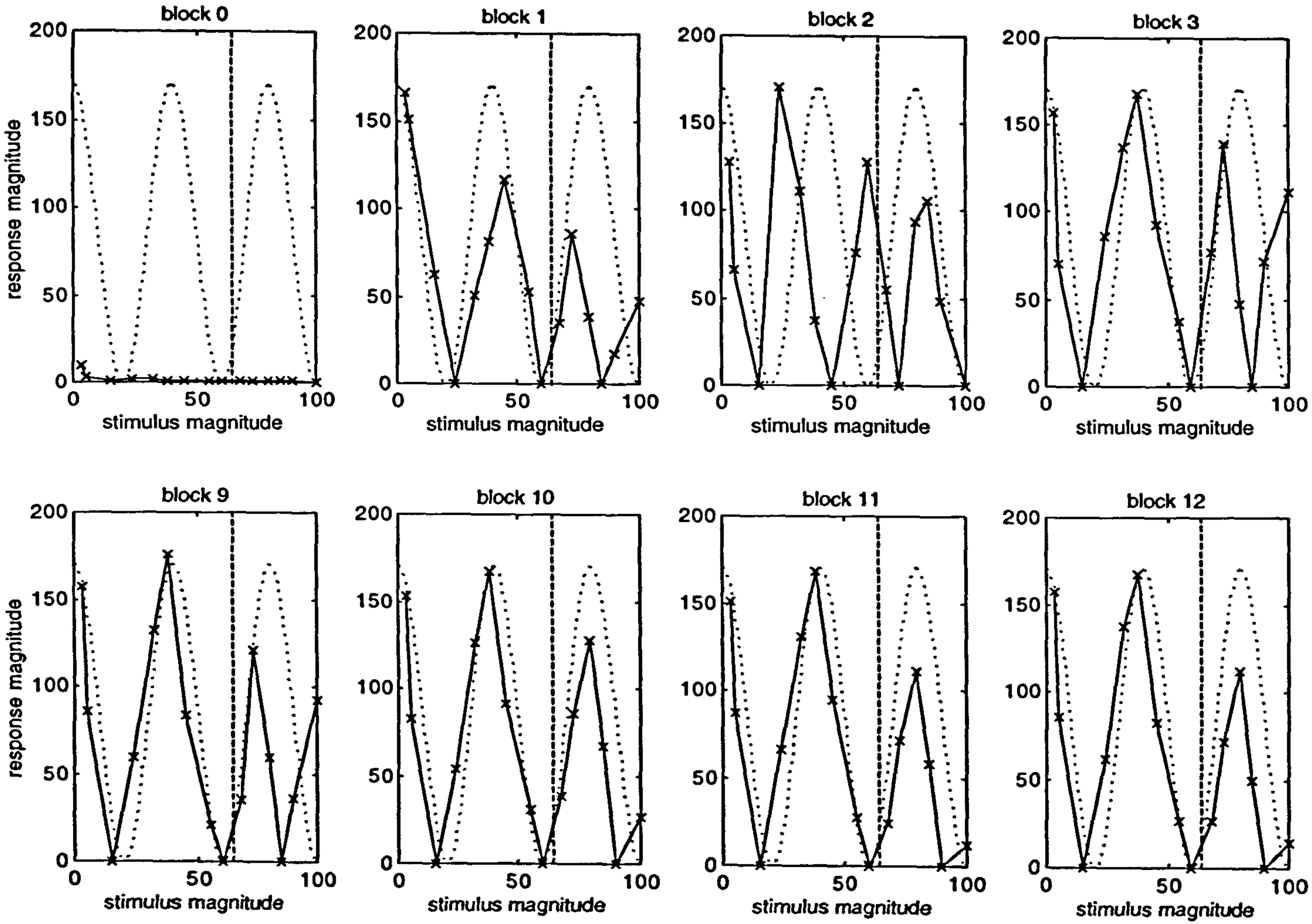


Figure 4.15 Participant 7's responses to the training and test data. Crosses to the left of the vertical dashed line are training points, while those to the left are testing points.

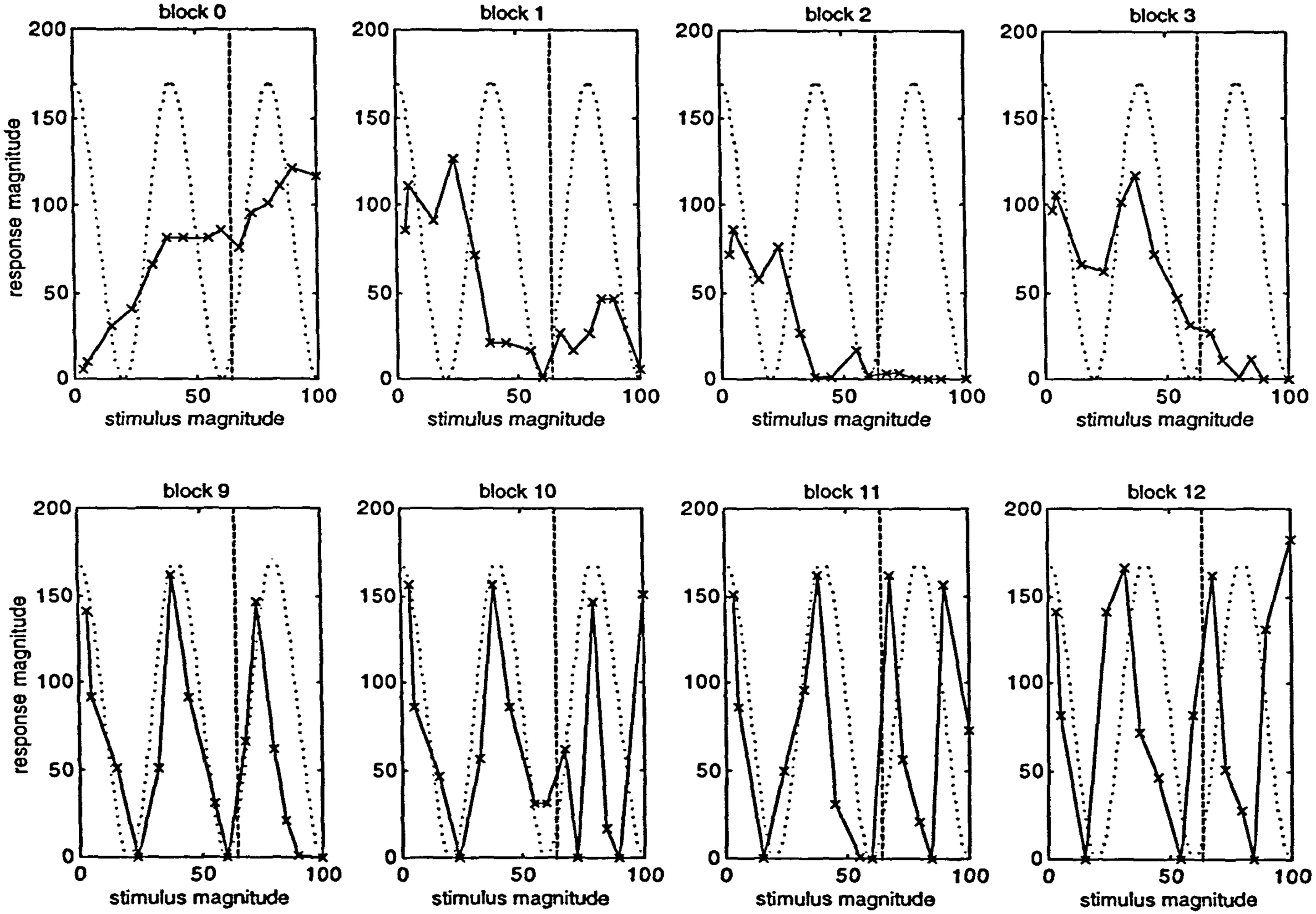


Figure 4.16 Participant 8's responses to the training and test data. Crosses to the left of the vertical dashed line are training points, while those to the right are testing points.

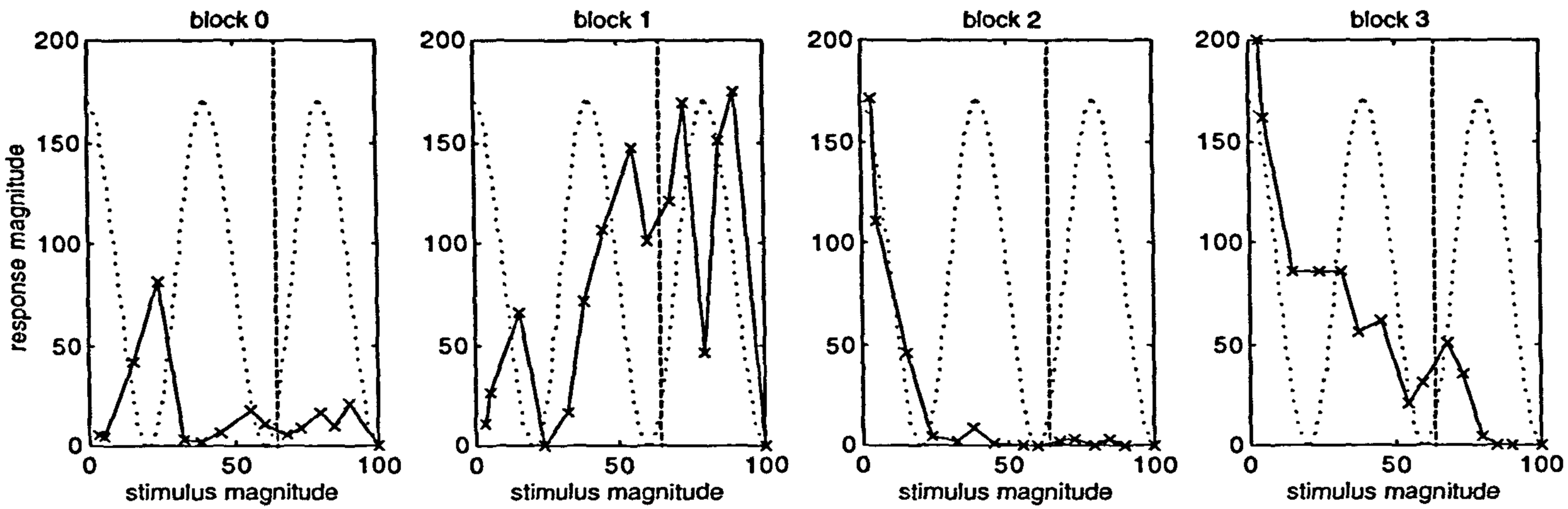


Figure 4.17 Participant 12's responses to the training and test data. Crosses to the left of the vertical dashed line are training points, while those to the right are testing points.

the cyclic function, albeit overestimating the frequency, like Participant 7. Finally, Participant 12 appears to have only slightly learnt the training data (if at all) and for the most part, seems to extrapolate on or around zero. This is in keeping with other participants early on in their training, but obviously very little can be drawn from Participant 12's responses.

As mentioned above, best-fit functions were determined on an individual participant and block basis. The first function to be fitted was the linear function, which was optimised by minimising the squared deviation from the participant's responses to the model's predictions. Table 4.2 shows the r^2_{adj} for the 13 testing blocks. The r^2_{adj} values reveal that most participants initially had good linear fits, but then moved away from this as training progressed, ending with very low r^2_{adj}'s. This pattern was confirmed by performing a repeated measure ANOVA on the mean absolute deviations from the lines of best fit, $F(12, 132) = 4.94$, Huynh-Feldt Epsilon = 0.51, $MSE = 155.43$, $p < 0.0005$. Given that EXAM was designed to account for extrapolation which was linear, evidence of a significant move away from linearity by participants must be taken as evidence against the model.

	Blocks												
	0	1	2	3	4	5	6	7	8	9	10	11	12
Pt 1	0.76	0.68	0.92	-0.23	-0.24	-0.14	-0.05	0.15	0.52	0.19	-0.24	0.78	0.26
Pt 2	0.85	0.21	-0.23	-0.25	0.53	0.09	0.11	-0.25	-0.25	-0.25	-0.21	0.20	0.1
Pt 3	0.93	-inf	-0.13	-0.25	-0.02	-0.10	0.11	-0.23	-0.23	-0.04	-0.24	-0.25	0.02
Pt 4	0.74	-0.25	0.68	0.17	0.8	0.56	0.83	0.31	0.11	0.85	-0.15	-0.17	-0.2
Pt 5	0.22	-0.11	-0.18	-0.19	-0.02	0.37	0.2	0.24	0.29	0.25	0.21	0.15	0.16
Pt 6	-0.22	-0.19	-0.09	0.24	-0.2	-0.12	0.51	-0.24	-0.27	0.00	0.31	-0.07	0.33
Pt 7	-0.85	-0.14	-0.21	-0.25	-0.25	-0.24	-0.04	-0.23	-0.01	-0.25	-0.02	-0.04	-0.03
Pt 8	0.74	-0.25	0.57	0.52	-inf	0.23	0.05	-0.22	0.3	0.5	-0.11	-0.22	-0.16
Pt 9	0.95	0.91	0.86	0.77	-inf	-0.03	-0.01	0.06	0.56	0.05	0.03	0.07	0.23
Pt 10	0.92	0.71	0.78	0.86	0.89	0.12	0.08	0.07	-0.25	0.11	0.3	0.17	0.26
Pt 11	0.8	-0.06	0.49	0.59	0.2	0.16	0.08	0.19	0.18	0.24	0.23	0.21	0.19
Pt 12	-0.25	0.01	0.18	0.64	-0.25	-0.10	-inf	-0.06	-0.16	-0.14	0.8	0.65	-0.14

Table 4.2 r^2 (adjusted) for the linear fit. Pt refers to the participant number.

	Blocks												
	0	1	2	3	4	5	6	7	8	9	10	11	12
Pt 1	-6928	-116.48	0.97	0.89	0.93	-0.35	0.84	0.92	0.84	0.90	0.95	0.92	0.99
Pt 2	-164.15	0.51	0.51	0.8	0.56	0.55	0.63	0.72	0.72	0.69	0.15	0.3	0.65
Pt 3	-0.83	-inf	-0.16	-0.25	-1.77	0.59	0.89	-2.05	0.73	0.9	0.81	0.6	0.82
Pt 4	0.83	-3.33	0.89	0.91	0.99	0.55	0.99	0.97	0.8	0.97	-8.01	-4.97	0.90
Pt 5	0.27	-0.47	-1.8	0.23	0.84	0.95	0.85	0.84	0.8	0.88	0.88	0.88	0.90
Pt 6	0.94	0.74	0.27	-2.21	0.42	0.28	0.81	0.81	0.84	0.77	0.99	0.95	0.91
Pt 7	-80777	-6.31	0	-0.08	-0.94	0.53	-0.37	0.11	-0.21	0.55	0.3	-0.53	-0.65
Pt 8	0.76	-0.25	-5673.8	-118.64	-inf	-20465	-0.11	0.5	0.32	-0.17	0.82	0.75	0.85
Pt 9	-8.73	-75.5	-29.53	-51.64	-inf	0.74	0.69	0.8	-3.68	0.31	0.23	0.62	0.68
Pt 10	-105.95	-149.1	-19.12	0.89	0.97	0.86	0.94	0.91	0.94	0.82	0.91	0.89	0.86
Pt 11	-3.48	0.81	0.98	0.98	0.93	0.89	0.91	0.92	0.92	0.93	0.92	0.91	0.91
Pt 12	-194.38	0.04	-5906.2	-21.06	-137.14	-29.5	-inf	-705.33	-16.68	-14.1	-46.26	0.72	-5.53

Table 4.3 r^2 (adjusted) for the cosine fit with 2 free parameters. Pt refers to the participant number.

Although the previous analysis certainly goes against any model which predicts linear extrapolation, we would hope to see responses move towards the target function if participants were picking up on the rule. As we mentioned before however, we cannot simply look at the deviations from the target function. Instead, we will show that participants' responses get closer to the cosine function with some free parameters. Informal examination of responses suggested that most participants were failing to optimise parameters b and c of the target function $y = 85 + 85*\cos((x+b)* \pi/c)$ at asymptote. We therefore estimated the two parameters for each block of each participant, and calculated the r^2_{adj}'s (shown in Table 4.3) and the deviations and from this function to the participants' responses. These deviations decreased significantly as the number of training trials increased, F (12, 132) = 2.19, Huynh-Feldt Epsilon = 0.54, MSE = 329.06, $p < 0.05$.

As a final test of the extrapolation performances, we ranked the responses to the six input values for each participant within each block. The ranks for the last six blocks (those blocks with asymptotic learning) were then averaged, and then ranked themselves. This produced one set of rankings for each participant, reflecting responses at the end of learning. Of the twelve participants, four produced rankings identical to what the target function (with no free parameters) would predict and four others displayed an inverted V shape, when plotted on an input magnitude by rank graph. The remaining four were participants 4,7,8 and 12. These rankings were consistent with our analysis of these participants above: Participant 4's extrapolation rankings had a V shape which could be interpreted as an underestimated frequency with linear extrapolation; Participant 7's

rankings demonstrated a tail indicative of overestimating the frequency and Participants 8 and 12 were essentially random, with Participant 8 perhaps in a V shape.

In summary, the results demonstrate that participants moved further away from a linear function as training progressed, but closer to the target function. On an individual participant basis, at least 9 out of 12 participants extrapolated at asymptote as if they were following the target function. Of the other three participants, Participant 4 seemed to have a linear response, and Participant's 8 and 12 have not learnt the training data sufficiently well for us to examine their extrapolation behaviour. In general, many participants seemed to start off extrapolating linearly, but then move towards a cyclic function as training progresses.

There are two criticisms which might be levelled against this experiment. First, it could be argued that we unfairly suggested cyclic extrapolation by giving participants a cyclic cover story. EXAM could therefore not be expected to predict our results because it is designed to be purely empirically driven, and not to incorporate the effects of prior knowledge. It is not clear however, whether the cover story affected the participants' performance. One would expect some kind of cyclic pattern in the pre-training test block (Block 0) if participants had taken in the cover story. Instead, most participants displayed approximately linear or random response patterns in this block, as shown by the linear r^2_{adj} values for this block (see Table 4.2). The exceptions to this are Participants 5 and 6, who showed quadratic and cyclic tendencies respectively. Furthermore, it

is difficult to imagine how cyclic instructions might be incorporated into the model, given its rule-based linear extrapolation. This issue is addressed further in the General Discussion, but the next experiment tests directly the effects of a cover story with a between-subject manipulation of the instructions.

The other difference between this experiment and others in function learning, is the sequential presentation of the training and testing points. This is a more serious criticism of the experiment, because it is very likely that participants are using their previous response as a basis for their current response. Presentation order would therefore crucially affect performance. Despite this, participants obviously feel that the cyclic nature of the training data continues in the extrapolation range – it would be perfectly possible to exhibit linear extrapolation and still treat the data as a time series. However, to answer this criticism, Experiment 3 used a random presentation order, allowing us to eliminate the hypothesis that the sequential ordering was responsible for the non-linear extrapolation.

4.2.3 Experiment 3

Experiment 2 involved a between subject manipulation of the instructions to assess the effect of telling participants that the function will be cyclic. In addition, input-output pairs were presented in a random order to prevent participants treating the function as a time series, which they did in the previous experiment.

Method

Participants

Thirty University of Warwick students were used as participants, none of whom had taken part in previous function learning experiments. They were paid £5 for completing the task, which took approximately an hour.

Design and Stimuli

There were 9 testing phases and 8 training phases, the extra testing phase occurring at the beginning of the experiment as a test of prior knowledge. Each training phase consisted of 8 different input-output pairs of points, each of those points being presented twice to make a total of 16 trials in any given training phase. Presentation order was random, although participants saw the complete set of distinct examples before any were repeated. The decision to repeat the points before having a test phase was made because it was felt that this was a

much more complex task than learning the points sequentially and therefore required more training time. The input-output pairs were again generated by the function $y = 85 + a \cdot \cos((x + b)\pi / c))$. In the test phase participants were presented with 20 inputs, ranging from 60-100 in gaps of 2.

The horizontal bar paradigm (as described in Experiment 1) was used to present and record the results.

A 2-level between subject design was used to test the effects of suggesting a cyclic function in the instructions, with 15 participants in each condition. These will be referred to as the group with Cyclic Instructions and Neutral Instructions respectively.

Procedure

Participants in the Cyclic Instructions condition received the cover story and stimulus labels given in the previous experiment, whereas the others received the following instructions with appropriate labels:

"In this experiment, we'd like you to learn a relationship between an input into a machine and an output from that machine. The machine will be taking in a substance called Drodine, performing some operations on that substance, and then finally producing a chemical called Sobacol.

You will be presented with a set of examples, each example consisting of the amount of Drodine that enters the machine, and the amount of Sobacol that is produced. Your task is to learn the relationship by a process of trial and error and the feedback provided by us."

The rest of the procedure was the same for all participants and is described in previous experiments.

Results and Discussion

Training Data

To assess the effects of the instructional manipulation and whether participants were learning the training items over blocks, a mixed 2 by 8 ANOVA was conducted on the mean absolute deviations of participants' responses from the target responses (MAE). The data were transformed using a cubic function to make the variances more homogenous. The ANOVA demonstrated a significant effect of Block, F (7, 196) = 24.61, Huynh-Feldt Epsilon = 0.59, MSE = 803649.25, $p < 0.0005$, but no effect of the instructions or of the interaction, all p's > 0.05. However, examining individual participant responses revealed that 4 of the 30 had extremely high errors at the end of the training phase. From the Cyclic Instructions condition, Participants 11, 13, and 8 had an equal or higher average MAE over the last two blocks compared to the first block. From the Neutral Instructions condition, Participant 23 had only a 10% reduction in MAE from first to last blocks. This is compared to a drop of 63 to 8 MAE for the

average of the other participants. Omitting the non-learning participants from the ANOVA showed no change in the effects. These participants will not be included in the data set in further analyses. For the remaining participants, all learning seems to be complete by Block 7.

Test Data

There were two goals of the analysis here. First, the effects of the instructions on the extrapolation responses were examined. Although there was no effect present in the training data, it might well be the case that the extrapolation range is more sensitive to this manipulation, given that responses to data outside the training range must involve prior knowledge in some way. Second, as in the last experiment, the form of the extrapolation responses are central to the predictions of the EXAM model. These two goals may of course be interrelated, by the effect of the instructions on the extrapolation. Furthermore, we can look at both of these factors from a group perspective or on an individual participant basis.

Group Analysis

The first analysis concerns whether or not participants get significantly closer to a linear function as learning takes place. This involved first estimating the best-fit straight line through the extrapolation data for each block of each participant. Then, for each block, the MAE's of the responses were calculated. This meant that there was one score per participant per block. These scores were then subjected to a mixed ANOVA with Block as a within factor and Instructions as a between participant factor. The results indicated a significant increase in the

deviation from a straight line as learning took place, F (8,192) = 2.80, MSE = 192.46, Huynh-Feldt Epsilon = 0.65, $p < 0.01$. There was no significant effect of the Instructions or the interaction, with all p's > 0.05. As before, a significant move away from linear extrapolation is contrary to EXAM's predictions.

A complementary analysis to the linear fits is to examine whether responses move closer to the cyclic curve which we used to generate the training data, as we established in Experiment 2. In this experiment, there wasn't as much variation in frequency or amplitude as we observed in the last experiment so we can simply take our dependent measure to be the participants' mean deviations from the target function's output responses (i.e. no free parameters are needed). Again, to homogenise variances a cubic transformation was used. The results indicate a significant decrease in the errors as learning proceeds, F (8,192) = 0.018, MSE = 547.19, Huynh-Feldt Epsilon = 0.58, $p<0.05$, although neither the main effect of Instructions nor the interaction were significant, p's > 0.05.

Individual Participant Analysis

To assess how each participant extrapolated, the last two blocks for each participant were combined and a best-fit function was found for each participant over the 40 data points. The functions we examined were the linear function and variations of the target function, $y = 85 + 85*\cos((x*\pi)/20)$. We examined models with at most 3 parameters, so that the most flexible model became $y = 85 + a*\cos((x+b)*\pi/c)$, where a, b, and c are the parameters to be optimised. There were therefore a total of 8 models: 3 from optimising each of a, b, or c; 3 from

altering each pair of *a*, *b* and *c*; 1 by optimising all three parameters at the same time and finally the linear model with two parameters. When a parameter wasn't being optimised, its value was set at the generating function's value, for example, when just the *c* parameter was being optimised, the values of *a* and *b* were set at 85 and 0 respectively. Tables 4.4 and 4.5 show the highest r^2_{adj} for each participant, together with the type of curve associated with the r^2_{adj} score, and whether or not that curve was nonmonotonic over the extrapolation range[4]. Monotonicity is included because it gives an indication of whether EXAM could account for a participant's responses. As demonstrated in the introduction, EXAM favours linear extrapolation and is unlikely to account even qualitatively for these participants.

These results show that there is considerable variability in the types of extrapolation responses made, with 19 out of 26 producing nonmonotonic responses, roughly evenly distributed over the two conditions of the Instructions factor. Important too, is the distribution of r^2_{adj} scores within the monotonic versus nonmonotonic categories; the monotonic curves tend to have a much lower score, indeed several participants have given essentially random responses. There are *some* participants who have extrapolated monotonically with a reasonably high r^2_{adj} e.g. Participant 22, but very few of them.

[4] To test whether these functions were monotonic, the optimised parameters for each participant and each block were inserted into the function and it was established whether or not this equation had a turning point over the extrapolation range (x = 60 to x = 100).

cyclic instructions			
participant	model	nonmonotonic	r^2(adj)
1	abc	1	0.75
2	bc	1	0.69
3	bc	1	0.76
4	abc	1	0.42
5	bc	0	0.21
6	abc	1	0.74
7	b	1	0.98
9	abc	1	0.57
10	a	1	0.97
12	linear	0	-0.03
14	bc	0	0.36
15	linear	0	0.50

Table 4.4 Model type, r^2 and monotonicity of best-fitting function over the last two blocks of testing data for participants in the Cyclic Instructions condition. 'linear' refers to the best-fit straight line with two parameters and the other models are combinations of the parameters from the $y = 85 + a*\cos((x+b)* \pi/c)$ model.

neutral instructions			
participant	model	nonmonotonic	r^2(adj)
16	c	1	0.13
17	bc	1	0.58
18	a	1	0.89
19	a	1	0.36
20	a	1	0.97
21	linear	0	0.72
22	abc	1	0.63
24	abc	1	0.87
25	linear	0	0.10
26	abc	1	0.10
27	linear	0	0.01
28	ab	1	0.55
29	a	1	0.14
30	a	1	0.78

Table 4.5 Model fitting results for the Neutral Instructions condition. Columns as above.

In summary, participants can extrapolate cyclically regardless of the presentation order of the stimuli or the instructions given to the participants. This is an important result because testing EXAM on the paradigm on which it was developed provides much more stringent evidence against it than using the design employed in Experiment 2. However, one criticism that could be made of the experiments is to say that participants are not really extrapolating the cyclic curve, but applying what they learnt in the lower input range into the extrapolation range. This could be achieved by performing a transformation on the input values in the extrapolation region to bring them back into the part of the space where output values are known. Extrapolation input values which, when transformed, did not appear in training could be given interpolated values from those that that did. This would imply that if participants were asked to extrapolated a curve which did not repeat itself, they would be unable to do it. Although a plausible explanation of what took place in this experiment, it is important to realise that EXAM still cannot reproduce these findings. The reason for this is that it has no mechanism of abstracting the period of the curve from the training data which would allow it to perform the transformation. In Chapter 5, the issue of abstraction is discussed in more detail.

4.3 Modelling

The experiments presented above suggest that EXAM is insufficient to model extrapolation behaviour. As a consequence, another model is developed here which provides a better account of the data. This model consists of two modules: a form of EXAM and a rule component. These two modules interact together via a mixing system so that at different times through the learning process, the weighting placed on the individual systems change, thus changing the extrapolation behaviour. The model will be referred as the Regression-Exemplar-Rule -Model (RERM). RERM was motivated by Erickson and Kruschke's (1998) model, ATRIUM, which was built to explain the interaction between rules and exemplars in a categorisation task. ATRIUM also consists of a rule component and an exemplar component which are linked together by a gating node. This gating node learns to allocate a different module to different areas of the space, depending on which module best classifies the past training stimuli in this space. The chief difference between this model and ATRIUM is the behaviour of the gating node, which will be discussed below.

4.3.1 RERM Description

Exemplar Module

Delosh *et al.* (1997) demonstrated that some kind of exemplar-based model with a linear extrapolation rule best explains past results on function learning.

Because of this, a form of EXAM will be incorporated into RERM. However, some changes are necessary, which simplify EXAM.

First, instead of dividing the input dimension into a large number of nodes, a representation assuming a single node for each training exemplar is used. Each node is then activated according to how close the stimulus pattern is to the training exemplar. There is no appreciable difference between these two forms of representation in terms of behaviour of the model. This approach is adopted because it is in keeping with the standard exemplar-based approach (Kruschke 1992; Nosofsky, 1986) and drastically reduces the number of weights to be trained. It also allows us to express the ALM as a standard radial basis function (RBF) network for regression (Bishop, 1995; Moody & Darken, 1989) and use its accompanying notation, described as follows.

In the RBF form of ALM, the data set consists of N input vectors x^n, together with corresponding targets t^n. The desired mapping, $h(x)$, is given by

$$h(x^n) = t^n \tag{11}$$

The RBF approach assumes a set of N basis functions, one for each point, which take the form $\phi(z)$ where $\phi(.)$ is some non-linear function and z is the absolute deviation, $\|x - x^n\|$. The output is then taken to be a linear sum of these basis functions

$$h(x) = \sum_{n} w_n \, \phi(z) \tag{12}$$

where the w_n are weights leading from each basis function to the output node. In the ALM's case, the basis takes the form of a Gaussian (see also Equation 5):

$$\phi(z) = \exp(-\lambda \cdot z^2) \tag{13}$$

with λ being the scaling parameter described in the introduction. The second difference between the exemplar component of RERM and EXAM involves changing the basis from Gaussian to linear, so that the functions simply calculate the absolute difference between them and the incoming stimuli. The activation of the basis function then becomes

$$\phi(z) = z \tag{14}$$

This means that interpolation is always piecewise linear, and extrapolation is also linear with the gradient determined by the weights leading from the basis functions. Although extrapolation is linear, it is not 'in the direction of the function', which Delosh *et al.* (1997) pointed out was necessary to explain past results in function learning. Because of this, a free parameter is introduced which aligns the gradient of extrapolation in the model with the responses of the participants. The output for the extrapolation is simply

$$h_{ext}(x) = g(x - x^{far}) + h(x^{far}) \tag{15}$$

where x corresponds to the range of test values in the extrapolation range, x^{far} is the furthest most furthest most x value of the training values, and g is a free parameter.

The reason for the change from EXAM to this exemplar-based system is that, when EXAM is presented with stimulus and target magnitudes from Experiment 3, it produces negative responses, which participants were prevented from entering. If EXAM is unable to predict the direction of extrapolation, as in this case, there seems little point in the response mechanism described by Equations 8-10. This in turn seems to render the scaling parameter and exponential transformation also unnecessary since, in EXAM, their primary role is to control the gradient of the extrapolation direction. These aspects of EXAM also control interpolation, but the results of Experiment 1 indicate that piecewise linear responses may be more appropriate than EXAM's response rule. When it comes to fitting the model, this change in response mechanism will only benefit a linear response model, so EXAM is not being unfairly treated. To summarise, a linear extrapolation mechanism is a core aspect of this module, but the Gaussian basis functions and the scaling parameter are unnecessary complications at this stage.

Rule-based Component

For these experiments, the rule module consists of a single cosine function. This can be thought of as a 1-dimensional network with one cosine function hidden unit and one output unit. The net has an adjustable first layer weight, first layer

bias, second layer weight and second layer bias. The output from the rule component is therefore:

$$h_r(x) = b_2 + w_2 \cos(w_1 x + b_1) \tag{16}$$

with the w's referring to the respective weights and likewise for the biases.

Mixing System

The output from the rule-based and exemplar components are combined using following equation

$$h_{er}(x) = \alpha h_e(x) + (1-\alpha) h_r(x) \tag{17}$$

where $h_{er}(x)$ is the output from the system as a whole; $h_e(x)$ is the output from the exemplar system, $h_r(x)$ is the output from the rule component, and α is a free parameter fitted from the training data. α thus controls the extent to which the overall response comes from the exemplar-component, or the 'representational attention' in Erickson and Kruschke's (1998) terms. Although there are similarities between the mixing system and the gating networks described by Jacobs, Jordan, Nowlan, and Hinton (1991) and Erickson and Kruschke (1998), one important difference is that RERM does not assume that different modules provide responses to different parts of the input space, as they do, but that at different stages of the *learning process* the exemplar component has different

contributions to the overall response. Thus, RERM can reproduce the finding that participants initially extrapolate linearly, but later on move attention across to the cosine rule and extrapolate nonmontonically.

4.3.2 Model Fitting

EXAM's extrapolation output responses are negative when fitted to the training data (they are outside the range of the response bars). Because participants were prevented from responding in this way, it would not be appropriate to compare RERM's fit with EXAM's. Instead, the exemplar component of RERM was used which assumes a linear extrapolation mechanism, but does not predict negative responses. This was felt to be in keeping with the principle of EXAM, but avoids the practical problems.

The models were fit to the data from individual participants because of the diversity of extrapolation patterns. However, only the model fits to Participants 7 and 22 (Experiment 3) are discussed in detail, because inferential analysis of the responses revealed that extrapolation was cyclic in many cases, thus demonstrating the inability of EXAM to explain all participants responses. The two participants were chosen because they display a range of behaviour which allows the model's flexibility to be demonstrated. A summary of RERM's fit to all of the data is presented at the end of the section.

This chapter has primarily been concerned with extrapolation behaviour and not with the learning algorithms participants use to estimate parameter values. However, it is interesting to look at how extrapolation behaviour differs at

different points in the learning process. To examine the model's learning behaviour then, the parameters will be optimised for each block, using the participant's responses to the training items and examine the deviation from the model to the participants extrapolation responses. Thus, there will be one extrapolation error score per block.

Parameters were estimated to minimise the sum of the squared error between the model's and the participants' responses. Weights and biases described by Equations 12 and 16 were optimised based on responses to the training items. The two free parameters, g and α from Equations 15 and 17 were based on participants' extrapolation responses. More specific modelling details are described below.

Exemplar-based component

On the training data, this component of RERM can reach zero error on any set of responses (as can ALM and EXAM) because it has an equal number of basis functions as training points. The weights can be found by finding the inverse of the matrix of activations, so that

$$\mathbf{w} = \Phi^{-1}\mathbf{y}_p \tag{18}$$

where $\mathbf{w}$ is the vector of weights, $\mathbf{y}_p$ the vector of participant responses and Φ is a square matrix with elements $\Phi_{nn'} = \phi\left(\left\|x^n - x^{n'}\right\|\right)$. For the extrapolation data,

the best fit straight line through the data is required, subject to the constraint that the line passes through the most extreme training point. This results in the following equations for the gradient, g, and intercept, c.

$$g = \frac{\sum_n x_n y_{np} - y_p^{far} \sum_n x_n}{\sum_n x_n^2 - x^{far} \sum_n x_n} \quad (19)$$

$$c = y_p^{far} - g x^{far} \quad (20)$$

where summation is taken across stimuli presented in extrapolation; x^{far} and y^{far} are the magnitude and response of the furthest training point.

The second columns of Tables 4.6 and 4.7 show the exemplar extrapolation errors for participants 7 and 22 respectively. Participant 7 starts off with linear extrapolation responses, but their RSS jumps after the first block and remains constant until the end. In contrast, Participant 22 is initially unsure, but then the SSE's become low towards the end of the experiment indicating the exemplar-model describes the data well.

Rule-based component

Fitting the cosine function from Equation 16 is a non-linear problem, so an iterative method is required. Optimisation was carried out using the Quasi-Newton algorithm (see Bishop 1995), which is an efficient supervised algorithm using error gradients that avoids many of the problems associated with gradient

descent. The algorithm was run 10 times with 10 different initial parameter values and selected those which fitted the training curve from these. The starting parameters included the weights in the cosine function used to generate the stimuli, and 9 random weight vectors.

Participant 7						
Block	alpha	linear module	rule-based module	RERM	linear vs RERM	rule-based vs RERM
1	1.00	4121.16	121324.96	4121.55	0.00	-391812.57
2	0.71	84433.31	111051.89	75206.40	0.00	0.00
3	0.00	102287.95	10903.15	10904.34	-6322.01	0.00
4	0.01	105084.67	6193.25	6207.30	-65678.25	0.00
5	0.00	116177.20	18800.02	18799.71	-803.13	0.00
6	0.00	107241.95	3569.38	3570.33	-415836.21	0.00
7	0.02	109718.95	1945.57	1854.80	-2557318.55	0.00
8	0.05	102205.87	1353.45	1020.98	-8599663.63	0.00
		731271.07	275141.67	121685.40	-3.92772E+20	0.00

Table 4.6 Modelling results for Participant 7. Column 2 shows the best fit alpha parameter, columns 3-5 show the SSE for the linear module, rule-based module, and RERM respectively, columns 6 and 7 show the chi-square values for the linear versus RERM test and rule module versus RERM test.

Participant 22						
Block	alpha	linear module	rule-based module	RERM	linear vs RERM	rule-based vs RERM
1	0.56	17456.81	24589.46	16003.75	0.00	0.00
2	0.21	79948.40	33866.29	26594.85	-5.22	0.00
3	1.00	66010.59	88814.09	66011.52	0.00	0.00
4	0.92	37915.06	110479.02	38269.35	0.00	-3.59
5	1.00	9490.21	97268.47	9489.69	0.00	-9321.70
6	1.00	33664.08	129681.57	33663.98	0.00	-39.81
7	0.88	5915.64	114544.92	4246.13	0.00	-301669.63
8	0.95	5325.56	106711.66	4797.46	0.00	-165009.36
		255726.36	705955.49	199076.72	0.00	-310410569

Table 4.7 Modelling results for Participant 22. Columns as above.

From the third column of Tables 4.6 and 4.7, it can be seen that Participant 7's RSS decreases to very low levels towards the end of training, whereas Participant 22's errors increase, as would be expected from the previous discussion of the exemplar component.

Mixing Component

Having fitted the two components, the only remaining aspect of RERM to model is the mixing system described by Equation 17. To do this, α is optimised from the extrapolation data, based on the output from the independent modules. Then, responses to the testing stimuli are retrieved from the model using Equation 17 with the best-fitting α value.

Results

The first column of Tables 4.6 and 4.7 show the α value of RERM for each block, demonstrating the extent to which the participants relied on the exemplar component to make their responses in extrapolation. What is noticeable from Participant 7 is the sudden drop in the α value at block 3, suggesting that some kind of switching mechanism is in operation, rather than a gradual change. On the other hand, Participant 22 shows mid values initially, but then approaches 1 as learning increases. These α mid-values allow us to see one of the advantages of using the mixing system described by Equation 17, namely that the error of

the overall system can be better than either of the two. The reduction in error, however, may be due to the increased flexibility of RERM to fit the noise.

Because of this, it is useful to consider whether the additional parameters of RERM are necessary, or whether a restricted model is more appropriate (either the exemplar component or the rule-based component). Likelihood ratio tests were carried out, therefore, between RERM and the linear component, and RERM and the rule-based component. χ^2 can be defined as

$$\chi^2 = -2\ln\left[\frac{RSS(general)}{RSS(restricted)}\right]^{n/2} \qquad (21)$$

where *RSS(general)* is the RSS for the general model (in this case RERM) and *RSS(restricted)* is RSS for the basic model (either the linear component or the rule-based component), and n is the number of data points. If the restricted model is correct, χ^2 has an asymptotic chi-square distribution with degrees of freedom equal to the number of restricted parameters (Lamberts, 1997, citing Borowiak, 1989). Tests were carried out on the RSS for each block, and the RSS summed over all blocks. The χ^2 scores for the model comparisons are displayed in the fifth and sixth columns of Tables 4.6 and 4.7, with asterisks indicating a reliable advantage for RERM. RERM has one more degree of freedom than the linear component, and two more than the rule module, which means that degrees of freedom are 1 and 2 for the likelihood ratio tests respectively. For Participant 7, RERM proves better than the linear component for blocks 3-8, and when

summing across all blocks. RERM only shows an advantage on block 1 when tested against the rule-module. For Participant 22, RERM is better than the linear module only in block 2, but better than the rule-module on blocks 5-8 and overall. These tests confirm what the value of the α parameter suggests; namely, that some participants respond with a linear strategy and then move on to a rule-based extrapolation, and some maintain linearity throughout. RERM therefore requires both a linear and a rule-based component to model these two subjects. Interestingly, no block showed that RERM was significantly better that *both* components, suggesting that RERM does not need to mix the two systems (as discussed above); either the linear *or* the rule-based module would be sufficient. This doesn't mean that the α parameter is redundant (without it, there would be no method of switching from one to module to the other, as Participant 7 did on Block 1), but that the processes might be better modelled by assuming the parameter can only take on a binary, rather than continuous value.

Tables 4.8 and 4.9 display the best-fitting α values for all of the participants. Values close to 1 indicate that extrapolation is linear, while those near 0 suggest that the rule component provides the best fit to the responses. As other analyses have made clear, there is considerable variation in extrapolation patterns. Nonetheless, as learning progresses, a general trend towards zero is apparent in the average α values shown at the bottom. This arises because many participants start off by emphasising the linear module, but then shift towards the rule-based module as more blocks are experienced.

Cyclic Instructions								
Participant	**Block 1**	**Block 2**	**Block 3**	**Block 4**	**Block 5**	**Block 6**	**Block 7**	**Block 8**
1	0.52	1.00	0.15	0.82	1.00	1.00	1.00	1.00
2	1.00	1.00	0.96	0.95	1.00	1.00	0.15	0.76
3	0.77	1.00	0.55	1.00	1.00	0.21	0.00	0.13
4	1.00	0.66	0.00	0.64	0.71	0.10	0.00	0.04
5	1.00	0.58	1.00	0.00	0.00	0.24	0.00	0.17
6	1.00	0.00	0.23	0.60	0.00	0.04	0.03	0.19
7	1.00	0.71	0.00	0.01	0.00	0.00	0.02	0.05
9	0.85	0.71	1.00	1.00	0.56	0.94	0.20	1.00
10	0.67	1.00	0.88	0.63	0.14	0.18	0.12	0.29
12	0.92	1.00	1.00	1.00	0.26	0.00	0.02	0.01
14	1.00	0.42	0.00	0.96	0.92	1.00	1.00	0.31
15	1.00	0.75	0.66	0.92	0.91	1.00	0.98	0.83
16	1.00	0.34	1.00	1.00	0.97	0.90	1.00	1.00
average	0.90	0.71	0.57	0.73	0.57	0.51	0.35	0.44

Table 4.8 α values for participants from Experiment 3 who received the Cyclic Instructions.

Neutral Instructions								
Participants	**Block 1**	**Block 2**	**Block 3**	**Block 4**	**Block 5**	**Block 6**	**Block 7**	**Block 8**
17	0.00	0.28	0.58	0.08	1.00	0.41	1.00	0.30
18	0.99	0.99	1.00	1.00	1.00	0.93	1.00	0.87
19	0.75	1.00	0.03	0.10	0.03	0.02	0.00	0.05
20	1.00	1.00	1.00	1.00	1.00	0.01	0.00	0.00
21	0.97	0.00	0.06	0.34	0.05	0.00	0.00	0.00
22	0.56	0.21	1.00	0.92	1.00	1.00	0.87	0.95
24	1.00	0.00	1.00	0.00	0.00	0.00	0.20	0.01
25	0.78	1.00	0.74	0.81	0.98	0.98	0.88	0.91
26	0.62	0.70	0.30	0.94	0.68	1.00	0.83	0.00
27	1.00	0.98	1.00	1.00	1.00	1.00	0.99	1.00
28	0.04	1.00	0.69	0.56	0.77	0.90	1.00	1.00
29	1.00	1.00	1.00	1.00	0.84	0.56	1.00	0.70
30	0.00	1.00	0.00	0.02	0.10	0.10	0.00	0.10
average	0.67	0.70	0.65	0.60	0.65	0.53	0.60	0.45

Table 4.9 α values for participants from Experiment 3 who received the Neutral Instructions.

The analysis of Participants 7 and 22 revealed very few intermediary values of the α parameter. This appears to be true generally of participants, as can be seen

from the histogram in Figure 4.18 - the distribution is heavily skewed towards the extremes of the range. Indeed, a Kolmogorov-Smirnoff test revealed significant departures from a uniform and a normal distribution, K-S(208) = 5.02, $p < 0.0001$ and K-S(208) = 3.00, $p < 0.0001$ respectively.

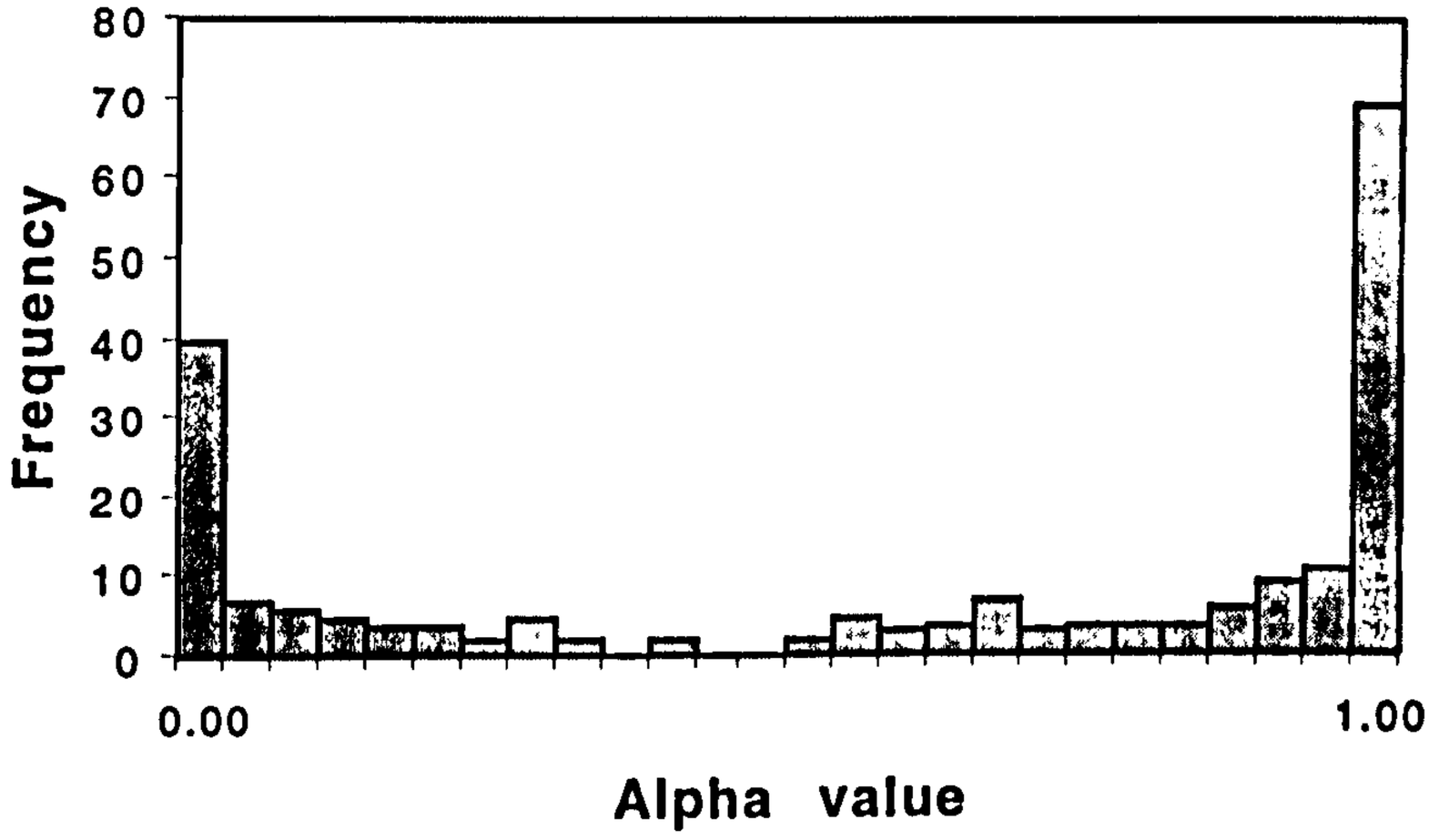


Figure 4.18 Distribution of α values based on 26 participants with 8 scores each from Experiment 3. Bin size is 0.04, except for the last category which is a count of all values scoring 1.00.

To summarise, a multiple component model with a mixing system like RERM is needed for two reasons. First, when modelling individual participants, it is necessary to account for the fact that some participants only extrapolate linearly at asymptote, such as Participant 22, whereas some extrapolate in line with the target function, like Participant 7. Secondly, a single participant is likely to apply different α values at different points in the learning sequence. For instance, Participant 7 starts off extrapolating linearly, but then changes to the cosine extrapolation as learning continues..

4.4 General Discussion

The three experiments presented in this chapter demonstrate several findings that challenge current theories of function learning. In the first experiment, it was shown that participants linearly interpolate when asked to generalise from three training points. This result suggests that participants might be fitting a series of straight lines to the data, and not, as Delosh *et al.* (1997) predict, a response mechanism based on Gaussian similarity functions and the linear response rule. The experiment also demonstrated an effect of the extremes of the response bars on participants' responses, which not only sheds doubt on some of Delosh *et al.*'s findings, but which will prove useful for other researchers using the paradigm.

Experiments 2 and 3 investigated whether participants would continue a cyclic pattern which they were presented with during training. Both experiments provided evidence that they do, despite variations in presentation order and instructions. Further, participants were shown to start off extrapolating linearly, then move into the more rule-based responses as learning progressed. This is the first demonstration of nonmonotonic extrapolation in the function learning domain and implies that participants have a far more complex mechanism than previously thought: Delosh *et al*'s explanation using a linear response rule is no longer adequate to explain participants' behaviour.

Because EXAM's were qualitatively different to the findings, another model was developed called the Regression-Exemplar-Rule-Model (RERM). This model has two components, a rule-based module and an exemplar-based module. The

exemplar-based component is very similar to EXAM, in that it assumes a linear extrapolation mechanism. The rule-based module can be considered a more parametric system, which can therefore extrapolate in non-monotonic ways. These two components are then combined to allow RERM to switch between a linear or a cyclic extrapolation as appropriate. The shift in extrapolation is controlled by a free parameter called the 'attention shift' parameter (see Equation 17), the value of which was shown to start off by emphasising the exemplar-model, then change to favour the rule-based component as learning progresses.

When RERM was fitted to the data, the exemplar component was taken to be a radial basis network with linear basis functions and an extra parameter to control the angle of the extrapolation gradient. The reason this system was used rather than EXAM, is that EXAM predicted responses out of the range of the methodology. This is clearly not always going to be the case and the question arises of which method would be more suitable in general. As discussed at the end of Experiment 1, the piece-wise linear solution would suffer with noisy data so it would have to be adapted to reduce the number of splines. Even once this is done however, it may not predict the correct direction of extrapolation (although it will always be linear). What might be needed is a response mechanism like EXAM's, but which is based on linear basis functions, instead of the Guassian. Future work could investigate which of these two models has the best fit to the data modelled by Delosh *et al.* (1997).

The attention parameter

One criticism that could be made of RERM is that it fails to predict the value of the α parameter from the training data, so that all the model's explanatory power is in this free parameter. One approach to this would be to minimise training error by allocating weight to the module which minimises training error the most. As RERM is currently construed however, fitting the α parameter in this way would mean that the exemplar component gets all the weight. This is because the it is sufficiently flexible that it can obtain zero training error on any set of data. Alternatively, learning algorithms could be implemented so that the exemplar component does not achieve zero training error on each block. This would mean that optimising on training error would be a possibility. However, the responses of the model would be entirely determined by the relative learning rates of the two components. For example, if the exemplar component learnt more quickly, then the extrapolation would be principally linear because the linear module predicted the training points better. In other words, using a free learning rate parameter adds nothing to the explanation of why participants shift from the exemplar modular to the rule module as learning progresses. Erickson and Kruschke (1998) faced a similar problem when modelling the interaction between rules and exemplars in a categorisation task. The difference between their model and RERM is that they had a gating node which learnt which part of the input space was best controlled by the different modules (see Chapter 2). They prevented the exemplar module responding to all parts of the space by setting the smoothing parameter to a relatively high value, thereby preventing it

from responding to some training items correctly. Consequently, the gating node could allocate attention on the basis of reducing training error. RERM could be augmented with a smoothing parameter (rather like EXAM with Gaussian basis functions), but, since the smoothing parameter is fitted on the basis of extrapolation responses, there would again be no additional explanatory power.

Yet another approach would be to base the α parameter on the error from the rule-based component only, so that when the error on the training data is reduced sufficiently, attention shifts to emphasise the rule-component. As we saw from Participant 7's responses, this seems a plausible explanation. However, the point at which a participant decides that the error is reduced 'sufficiently', can only be determined from the extrapolation responses, and so we are back to where we started. Finally, the exemplar-component could be changed so that the number of basis functions were reduced from N, the number of training points, to a smaller number, which would mean that the exemplar component would be unable to achieve zero error. This is the exact situation that was discussed above and at the end of Experiment 1. If this were the case, a possible hypothesis is that α might increase in proportion to the reduction in exemplar error gradient (with respect to presentations of stimuli). The only drawback here is that the number of basis functions now needs to be estimated, which again would be best optimised through examination of the extrapolation data.

It seems that the behaviour of the attention parameter cannot be determined by modelling the present experiments. To understand why, consider what α does in

a more statistical sense: α switches the system from one which is highly data driven, to one which is far less flexible and controlled by its background assumptions. In other words, α controls the smoothness of the regression solution, much like the number of hidden units in a neural network, or the λ parameter in the ALM and EXAM, or the number of splines used to fit a function. As Chapter 2 discussed in detail, the smoothing parameter has to be determined in part by background assumptions. Future work needs to investigate what kind of information influences smoothing; the effects on extrapolation responses is one method of observing these.

Although EXAM and other non-parametric models provide a poor account of the findings presented here, the parametric models discussed in the introduction (e.g. Brehmer, 1974; Koh & Meyer, 1991) would certainly provide a good fit to the some of the asymptotic extrapolation responses. Indeed, the rule module of RERM is simply a parameterised cyclic function. On the other hand, it is clear that some participants continue to extrapolate linearly at the end of training (see Tables 4.4 and 4.5 for examples) which would require a model to have a linear extrapolation component. Furthermore, several experimenters have demonstrated that participants start off with an initial expectation of linear relationships (Brehmer *et al.*, 1974; Byun, 1995; Naylor & Clark, 1968) and that extrapolation with quadratic training points is linear (Delosh *et al.*, 1997). Taken together, these results imply that only a dual component model like RERM can provide a sufficiently flexible account of the data. The question of why participants extrapolate linearly in some experiments and continue the function is

others remains unresolved, but perhaps the answer to this is related to the question of how people decide on the appropriate attention parameter setting.

Although the modelling demonstrated that both components of RERM were needed, very little evidence was presented that both were needed *in the same block.* This lead to the suggestion that the data might be better modelled by assuming that the attention parameter should be binary, rather than continuous. The implication of this is that the learning procedure might be best thought of as one that adapts its architecture serially, rather than one that optimises both components in parallel and selects the most appropriate to make the response. Of course, the account still provides no guidance as to why participants make the change from exemplar to rule-based module, but the distinction might become important when comparing RERM with other models or investigating the specific learning algorithms that people use.

Sequence effects

In Experiment 2, participants observed the training data in a sequential order, whereas in Experiments 1 and 3, the data was in a random order. It is worthwhile discussing the differences which may arise between these two presentation methods and the implications for the models. There have been very few studies comparing learning in the two presentation orders, but Byun (1995) carried out some studies showing that learning a quadratic function, a linear function and a cyclic function was easier if participants were presented with the data sequentially than if they were presented in a random order (although she didn't test extrapolation responses). Byun suggested that benefits arise because

of a reduced memory load that allowed participants to dedicate more recourses to discovering a rule in the sequential presentation order. Presumably, the reduced memory load comes from participants learning to predict the output responses on the basis of the preceding output value and treating the situation as a time series. This implies that they had two sources of information on which to base their responses: the time series and the input dimension. However, contrary to Byun's analysis, the order effects are not evidence of a rule-testing; it is only necessary to postulate that participants in one condition had more information available to them than in other to explain differences in accuracy.

In their present forms, neither EXAM nor RERM are appropriate for modelling the effects of presentation order. The reason is that they are both feed-forward models which assume that trials are drawn independently of each other, whereas what is needed is an algorithm capable of acting on the relationship between trials. For example, a recurrent network (e.g. Cleeremans & McClelland, 1991) would be able to learn to associate an output value O_1 which occurred at time t_1, with an output value O_2 which occurred at t_2, thereby producing O_2 when presented with O_1. The feed-forward models however, simply have no way of encoding the time dimension. Further modelling work might investigate the extent to which participants rely on the time series data, and to what extent they focus on the relationship between input dimension and output dimension. This could be achieved by using a mixture of experts architecture with the different experts corresponding to different sources of information (in a similar way to Baywatch, Chapter 2, and RERM, above).

An alternative explanation for the order effects is that the sequential presentation facilitated the optimisation algorithm of a feed-forward network. This idea was put forward by Busemeyer *et al.* (1997), who carried out simulations with training data in a random order or in a 'systematic' order (presumably sequential). They used EXAM with a Hebbian learning rule and initial weights set to reproduce a linear mapping. Busemeyer *et al.* report lower mean absolute errors for the systematic condition when learning a negative linear function and a quadratic. However, Busemeyer *et al* don't offer any explanation for why they get these effects and there are several reasons not to generalise from these simulations. First, the most likely reason for the effects is that the model was able to avoid local minima when certain presentation patterns were combined with certain stimulus sets. Busemeyer *et al.* only used one stimulus set per function, which means that it is impossible say whether the advantage gained for sequential presentations is true for all functions of that type, or just for that particular data set. Secondly, they only report results using one learning rate and there is no indication of whether the networks were trained to asymptote or not. Given that the global minima can be found analytically, a low enough learning rate would imply no difference between them at asymptote. Finally, Byun (1995) conducted similar simulations and concluded that there was no difference between random and sequential orders for EXAM (cf. Busemeyer, Byun, Delosh, and McDaniel, 1997). These mixed conclusions imply that more work is required to establish why there is a facilitation for participants learning sequentially presented training data.

4.5 Conclusions

Delosh *et al.* (1997) described extrapolation as the *sine qua non* for abstraction in function learning; their point being that with participants only showing linear patterns of extrapolation regardless of the pattern in the training data, there was insufficient evidence to say that they were abstracting 'functions'. The principle finding from the experiments presented here is that nonmonotonic patterns of responses are possible given the right training data. Thus, a model which doesn't abstract, such as Delosh *et al.*'s EXAM, needs to be substantially modified if it is to be a generic model of function learning.

This chapter has also highlighted the relevance of prior knowledge in perceptual domains. This has been achieved in two ways. First, the experiments have demonstrated that people chose to apply their background knowledge in their responses: despite having the opportunity to apply a non-parametric model, they chose to fit a solution which is far less flexible, but is known from past experience. Secondly, the modelling work has emphasised the difficulty of choosing the appropriate smoothing level when fitting a function. Arguments in the General Discussion made clear that participants weren't simply fitting the solution which minimised training error; they were applying some form of prior knowledge to make the choice about which function fit.

Chapter 5

In the previous chapter, Delosh, Busemeyer, and McDaniel's (1997) models (ALM, and the extrapolation version, EXAM) were shown to be incapable of explaining nonmonotonic extrapolation in a function learning task. A further problem with their model, and indeed all non-parametric models, is that they have only a limited way of representing a *function*, as opposed to a collection of individual exemplars. Consequently, the models cannot benefit from knowledge about the function which generated the data. The two experiments presented here test whether human participants can benefit from such knowledge.

At the end of learning, EXAM has learnt a reasonable approximation to the true mapping between input and output values that it has been trained on. In one sense then, EXAM has learnt a function describing that mapping. However, the knowledge embodied in the model seems very much tied to the current stimulus values. If another learning situation arose where it was known that the function was of the same type but with different parameter values, could EXAM make use of this information? To answer this question, it is first necessary to examine what is meant by the claim that two mappings are 'of the same type'. The function learning literature from psychology (e.g. Brehmer, 1974; Busemeyer, Byun, Delosh & McDaniel, 1995; Sawyer, 1991; Sniezek, 1986) has implicitly assumed that mappings are the same type if they are of the same order polynomial in x (the input dimension). For example, the claim that the linearly increasing functions are learned faster than non-linearly increasing functions

(Busemeyer *et al.*, p. 409) implies that the group of 1st order polynomials form a coherent psychological group.

The original question can now been seen as asking whether EXAM could make use of the knowledge that the true mapping will be a k^{th} order polynomial (where k is specified). In terms of the information value that this knowledge provides (see Chapter 2), the answer must be in the negative; EXAM is unable to restrict the range of allowable mappings to polynomials of up to k. This is not to say that it cannot restrict the range of possible solutions – specifying a λ value before learning achieves this – but rather to say that the model is unable to restrict them to the appropriate group. There are two reasons for this. First, the similarity functions of EXAM are exponential, whereas those of the polynomial described above are increasing powers of x. Secondly, EXAM assumes one basis function per training point, while the polynomial makes no such assumptions. If EXAM had the freedom to specify the type and number of basis functions to which it fitted a solution, then it would be perfectly capable of making use of abstract information such as that the mapping for the to-be-learnt task was quadratic in x, say.

This appears to prevent EXAM from reproducing rate of acquisition effects, such as the finding that monotonic functions are learnt faster than non-monotonic functions (e.g. Brehmer, 1974), or that cue labels that suggest the correct functional relationship facilitate performance (Byun, 1995; Sniezek, 1986). This is because if the model does not have a method of representing these functions, then, *ipso facto*, it cannot explain why some classes of functions are learnt more

quickly than others. However, Busemeyer *et al.* (1997) suggest that EXAM can explain the apparent function-guided behaviour by assuming that the algorithm is facilitated by knowledge which reduces the *complexity* of the task (again, see Chapter 2). Specifically, they maintain that advantages are obtained by setting initial weight configurations so that an appropriate mapping would be present before any learning took place. For example, setting the weights from input to output to 1 for equal values of input and output and zero otherwise would represent the identity $y = x$ and explain why a positive linear function would be learnt before a quadratic (Byun, 1995). There are several drawbacks to this approach however, as discussed in Chapter 2, Section 2.3.1. First, the knowledge can be completely surpassed by the data - after a long period of learning, the goal of minimising the training error will completely wipe out the initial weight configuration, and the prior knowledge. If the learning examples contain noise, the final weight solution will reflect that noise, thereby removing one of the benefits of the prior knowledge. Secondly, the scale difference between the generating function and the initialised function has far more of an effect than the mathematical similarity of the two functions. For instance, if the initial weights are set to represent $y = x$ varying over an output domain of 0 to 100, whereas the generating function is $y = x + 1000$, the weights will require large amounts of adjusting, regardless of the fact that both functions are linear. Busemeyer *et al.* (1997) implicitly acknowledge this last point later in their chapter by referring to this method of inserting prior knowledge as "a rather crude approximation" (p.425) and suggesting a *proportional prior-knowledge* assumption instead. This method maps the minimum cue value onto the minimum criterion value, the maximum cue value onto the maximum criterion

value, and intermediate stimuli are mapped proportionally. In other words, a straight line is initially mapped between minimum and maximum stimuli points. This approach avoids the scaling problem, but other the problems inherent in representing prior knowledge as initial weight configurations still remain. It could also be said that knowing the proportional prior knowledge rules constitute the representation of abstract functions in a polynomial manner. It therefore seems a waste of resources not to fit them parametrically to the data.

Even though there are some theoretical concerns with using initial weight configurations to represent prior knowledge, there is still a need to test EXAM's explanation of prior knowledge empirically. The first experiment presented here is a transfer task with two learning stages. The experiment tests whether learning the same functional relationship in both stages facilitates performance. An initial weight configuration for the second phase of learning can be achieved by optimising the ALM to produce the mapping from the first phase (EXAM is not needed because the experiment is not concerned with extrapolation). The ALM's performance on the second phase can then be compared with those from the participants.

5.1 Experiment 1

Participants in Experiment 1 were taught one of three functions in the first phase of the experiment (Stage 1): a positive linear (PL); negative linear (NL) or quadratic function (Q). In the second phase (Stage 2), all participants were taught a quadratic function but one with different parameter values to that encountered in Stage 1. The stimulus values are shown in Figure 5.1.

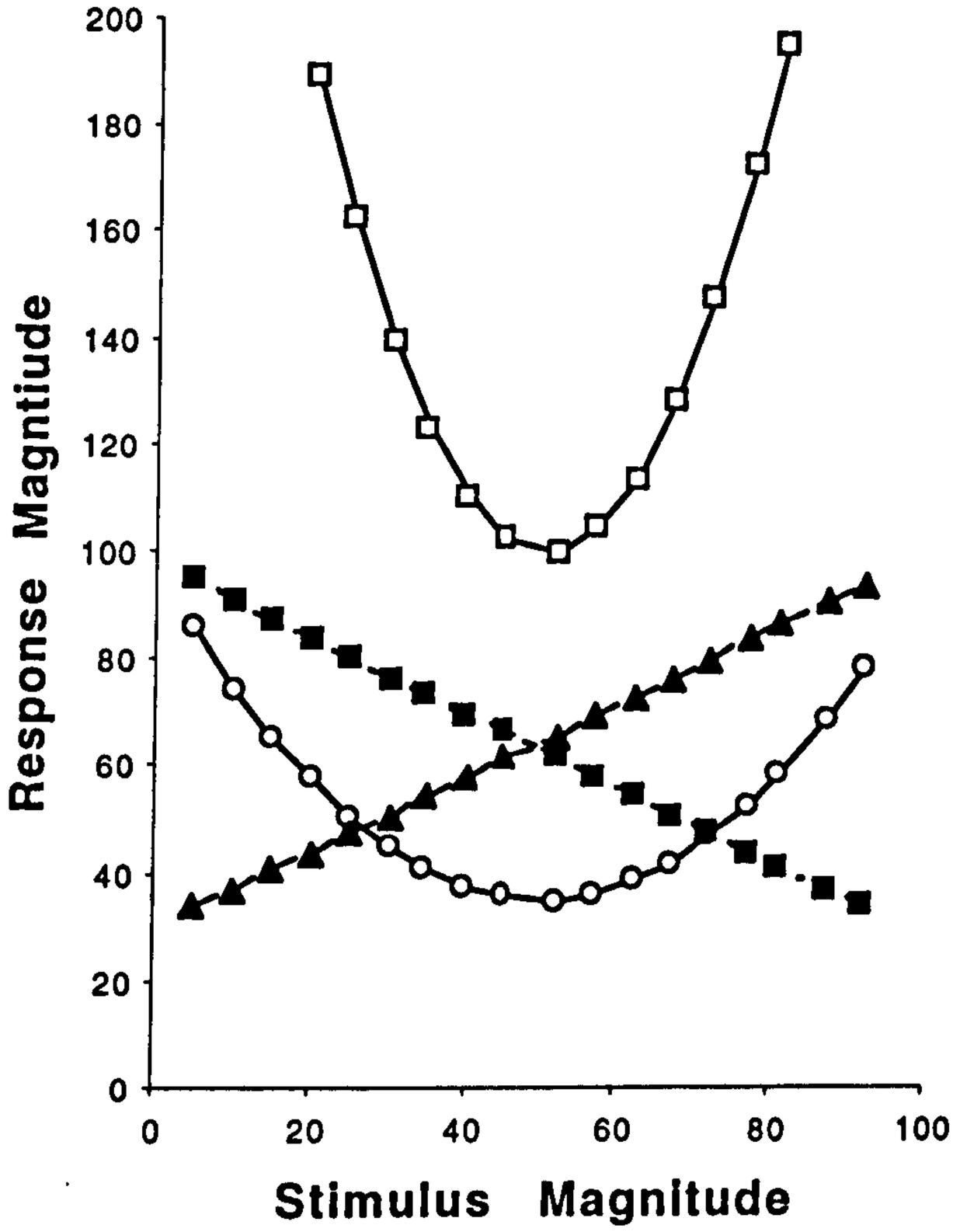


Figure 5.1 Stimulus and response magnitudes for three, between-subject Stage 1 conditions, that is lower quadratic, positive and negative linear functions. Above these is the Stage 2 function, the higher quadratic, which all participants learn.

As well as testing the ALM's predictions (which are described below), Experiment 1 was also designed to examine hypotheses derived from a more parametric approach. In particular, work by Flannagan, Fried and Holyoak (1986) on distributional category expectations is relevant. Flannagan *et al.* showed first that participants learn a normal distribution more quickly than other distributions. Next, they demonstrated that when participants are exposed to a multi-modal or skewed distribution in a category learning task, they were

facilitated in learning a multi-modal distribution in a second category learning task, relative to a group which learnt a normal distribution in the first task. Their interpretation of the findings was that the normal distribution was the 'default', but that participants could be placed into a state of readiness for a non-normal distribution by a preceding task. This readiness was not found to be distribution specific however, because the learning of the multi-modal distribution in the second task was facilitated by the prior learning of a skewed distribution, as well as a multi-modal distribution. By making the positive linear mapping in function learning analogous to the default normal distribution, the question can be asked whether participants become primed to expect a function which is not positive linear, as Flannagan *et al.* might predict, or whether the facilitation is function specific. These hypotheses can be tested by examining the performance on the Stage 2 learning task. Facilitation from both negative and quadratic groups would imply non- positive linear expectations, while a detriment to learning from the negative linear group and facilitation for the quadratic group would imply a function-specific account.

ALM simulations and predictions

Predictions were derived from the ALM by carrying out three simulations representing the three between-subject conditions of the experiment, that is, whether participants learnt a positive linear function, a negative linear function or a quadratic. The procedure for the simulations was as follows. First, the ALM was optimised to produce the appropriate mapping using the stimuli values shown in Figure 5.1. Optimisation was achieved by linear algebra and produced

zero error on the training items. Next, the model was trained on the stimuli values shown by the upper quadratic function in Figure 5.1, with the weight solution from Stage 1 as the initial configuration. Training in Stage 2 was done using gradient descent on the error (with learning rate at 0.05), to enable learning curves to be generated.

Before providing the results of the simulations, several complications need addressing. As Figure 5.1 shows, there are fewer training values in Stage 2 than in the Stage 1, and therefore fewer basis functions needed in the ALM. However, if the unnecessary basis functions are removed, not only is the impact of the prior knowledge reduced, but the weights from the other nodes are not correctly optimised to produce the mapping from Stage 1. For this reason, the ALM contained basis functions left over from Stage 1, which do not receive training in Stage 2. This allowed the weight matrix from Stage 1 to be used directly in the Stage 2 simulations.

A further problem concerns the smoothing parameter values, λ in the ALM. The value of this parameter alters the generalisation gradient from the basis functions and consequently the final weight matrix. This parameter is usually estimated as a free parameter from extrapolation responses made by participants but, because this experiment is concerned purely with training responses, this is not possible. A solution is to estimate it based on past research. Figure 5.2 shows ALM performances on training data and interpolation data optimised on the lower quadratic Stage 1 function. Generalisation patterns are shown with $\lambda = 0.5$ and $\lambda = 0.005$. Note that interpolation responses are approximately as accurate as

training data for $\lambda = 0.005$, but not for $\lambda = 0.5$. Since Koh and Meyer (1991) and Delosh *et al.* (1997) have shown no difference between performance on interpolation and training data in empirical experiments, λ was set at 0.005 for the simulations.

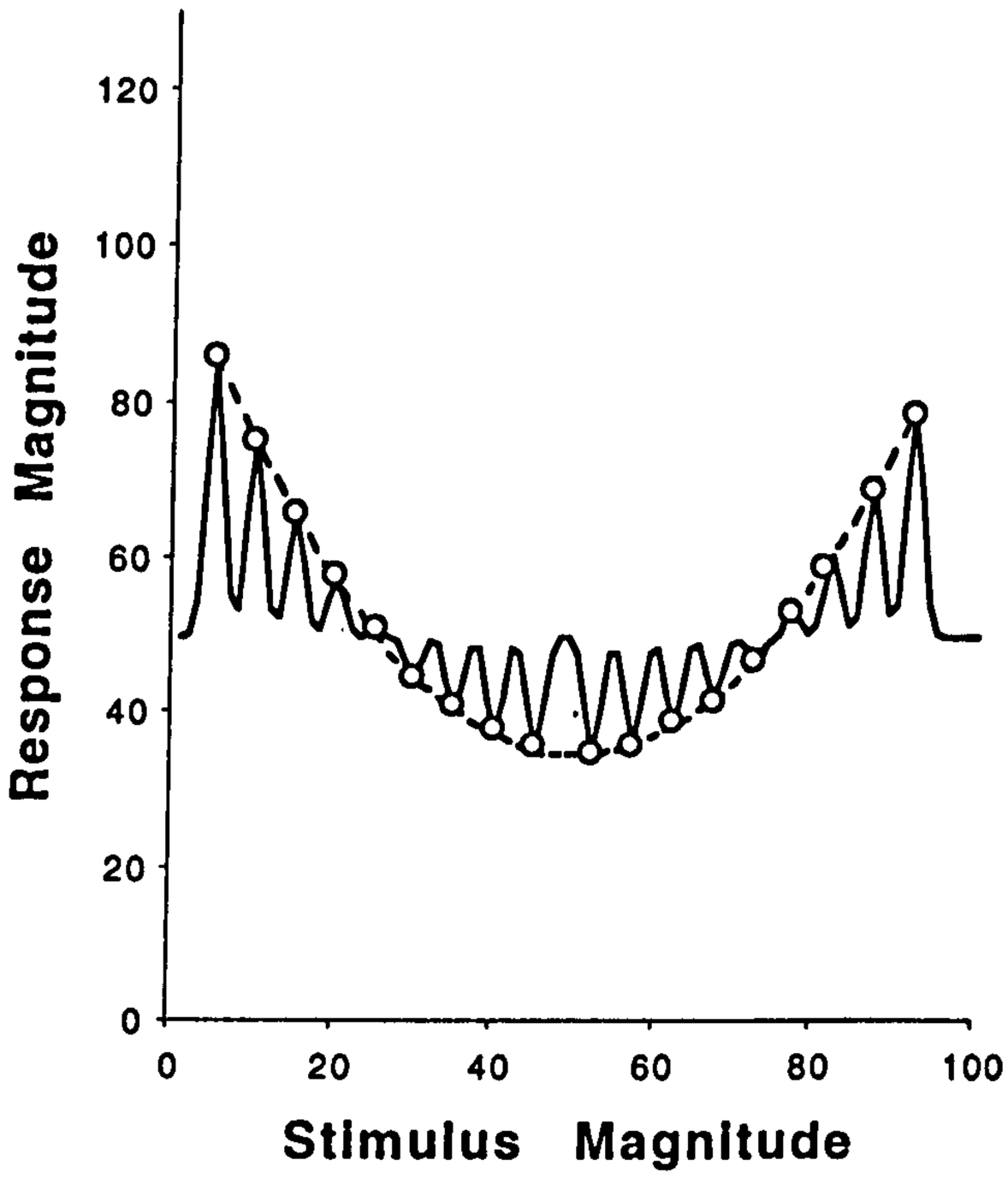


Figure 5.2 Response predictions from the ALM after being trained on the lower quadratic stimuli (see Figure 5.1). The circles correspond to training stimuli, the jagged solid line are generalisation responses with $\lambda = 0.5$, and the smooth dashed line are generalisation responses with $\lambda = 0.005$.

Figure 5.3 displays the results for simulations of ALM on Stage 2 stimuli, that is the upper quadratic shown in Figure 5.1. The different lines represent different

initial weight matrices, corresponding to the weight solutions for ALM to represent either the positive linear, negative linear, or quadratic functions. The key predictions that can be drawn from these simulations is that participants would be expected to show a clear disadvantage of having the quadratic initial weight matrix, as evidence by the far higher MAE in early blocks, and that there is no observable difference between positive and negative linear conditions. Of course, the results of the simulations arise not because of any functional similarity between the linear mappings and the Stage 2 quadratic, but because the linear initial weight matrices are closer to the final Stage 2 matrix than the Stage 1 quadratic weights are. This seems a fragile result, in the sense that different stimuli may produce different findings, but this is the only way of generating predictions from the ALM – it cannot represent a function as distinct from individual stimuli mappings.

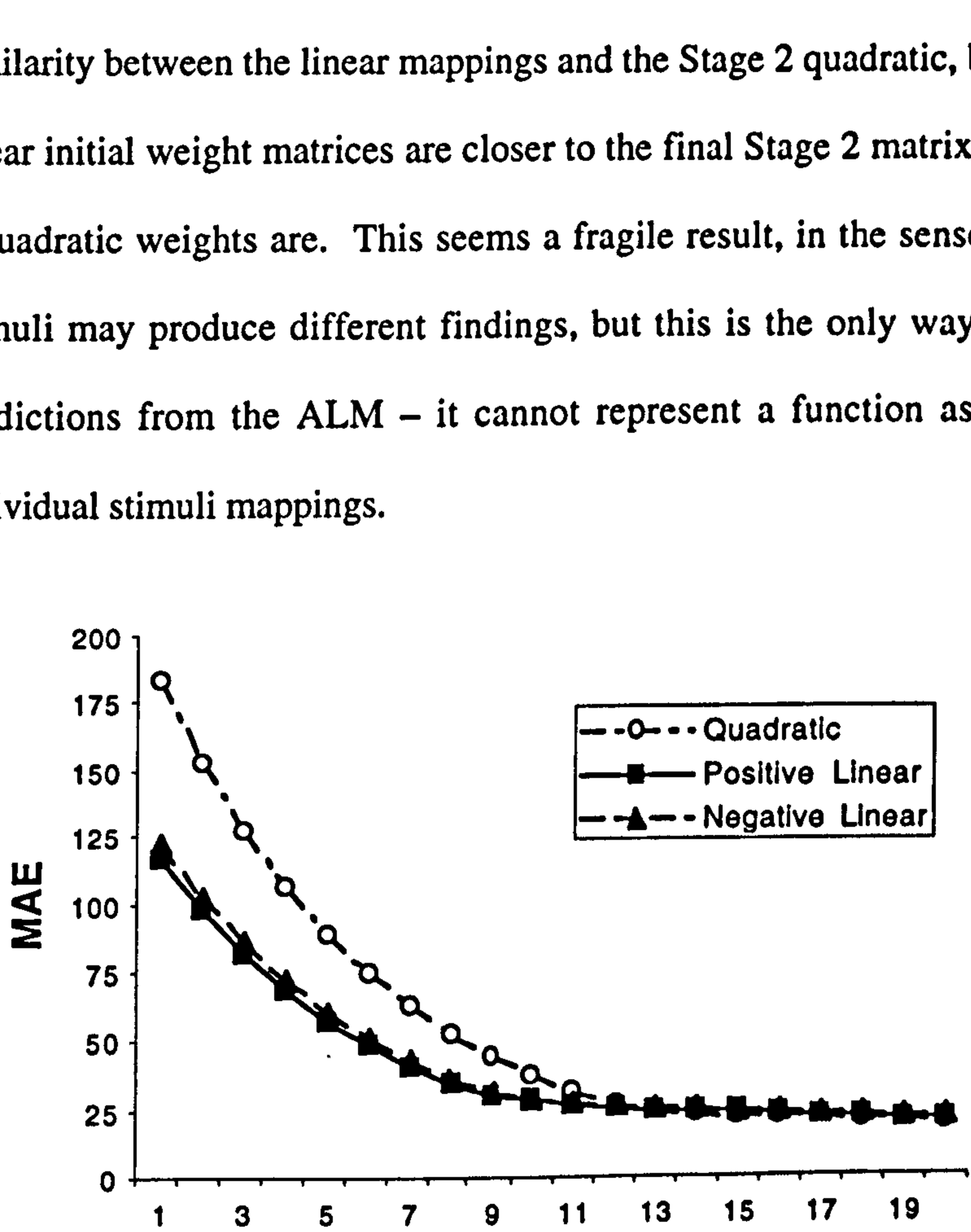


Figure 5.3 Mean absolute error of ALM responses from target responses on Stage 2. The three curves refer to simulations initialised with weight configurations specified in the legend.

Method

Participants

47 University of Warwick students participated and were paid £5 for the experiment, which took approximately an hour. Two participants were removed from the analysis (see Results section) leaving 15 remaining in each condition.

Design and Stimuli

In Stage 1, participants learnt either a positive linear (PL), negative linear (PL) or quadratic function (Q). Participants received training blocks, where they received feedback, and test blocks where they didn't receive feedback. There were 9 training blocks, each consisting of 18 input-output examples from the appropriate function. Within each block, all points were presented in a random order, one at a time. All three conditions received the same x-values, but different y-values. The equation relating the input to the output for the PL condition was $y = 0.7x + 30$, for the NL condition it was $y = -0.7x + 98$, and for the Q condition $y = 35 + (x - 40)^2 / 40$. The x-values for all 3 conditions ranged from x=5 to x=92. There were 2 testing blocks of 15 x-values ranging from x=8 to x=80. These x-values were a mixture of values they had seen in the training phase, and interpolation values. They were not provided with feedback. One testing block was presented after the 5th training block and the other one after the 9th. The test phases were built into the experiment to encourage participants to extract the underlying function.

In Stage 2, all participants were trained on a quadratic curve with the equation relating the input to the output being $y = 100 + (x - 50)^2 / 50$. This is the upper quadratic displayed in Figure 5.1. Again, there were both training and testing blocks. Participants were exposed to 9 training blocks, where they saw 13 input-output pairs presented randomly within each block. The range varied between x=20 and x=82. The change in range from Stage 1 to Stage 2 was because at more extreme x-values, the correct y-values were outside the range of the experimental equipment (i.e. over y=200). There were three testing blocks of 12 x-values, ranging from x=22 to x=80, which tested both training and interpolation values. Testing blocks took place after the 3rd, 6th and 9th training blocks.

Stimuli were presented graphically in the form of horizontal bars, as described in the previous chapter. In Stage 1, the input and output were red on a blue background, and in Stage 2 they were green on a blue background.

Procedure

Participants were told that they were going to be learning the relationship between the amount of a drug which enters a system and the level of arousal it causes. They were then instructed on how the quantities of drug and arousal would be represented, and how the they would receive feedback (see the previous chapter for details). After they had completed Stage 1, they were presented with the following instructions on the screen:

"In the second half of the experiment, you will learn now learn the relationship between a different drug, called Soromine, and arousal levels. Bizacol and Soromine are different drugs, but they affect the arousal level in similar ways."

The instructions were designed to encourage participants to use any information in Stage 2 that they had abstracted from the Stage 1 learning phase.

Results

Two analyses were conducted. The first used a relatively complete set of participants' responses, whereas the second used only the data from the 6 most accurate participants in each condition. In describing the trimmed analysis, only the results which differ to those of the full treatment are reported. All analyses used data which were transformed by a cube root to homogenise variances.

Full analysis

Two participants were removed from the analysis because their mean absolute error (MAE) was higher in the final block of learning than in the first. In addition to this, 34 responses were removed (out of a total of 15525) because their responses were 0, indicating that participants simply pressed the RETURN key to store the response without taking any notice of the screen. Byun (1995) and Delosh *et al.* (1997) removed responses for the same reason. Furthermore,

all the associated reaction times were below 500 msecs, which is an insufficient length of time to perceive and consider the stimuli bars.

Training Phase

Figures 5.4 and 5.5 show the learning curves for the three conditions in Stages 1 and 2 respectively. Stages 1 and 2 are analysed with separate ANOVA's on MAE scores for each block. In Stage 1, all function conditions show a reduced MAE as more blocks are experienced, as indicated by the significant main effect of blocks, $F(8,336) = 108.53$, Hunyh-Feldt Epsilon $= 0.93$, $MSE = 0.05$, $p < 0.0005$, and the non significant interaction between block and function condition, $p > 0.1$. The main effect of function condition was significant, $F(2,42) = 23.65$, $MSE = 0.66$, $p < 0.0005$. ANOVA's on the individual pairs of conditions revealed that participants who learnt the Positive Linear function had consistently lower MAE's than those who learnt the Negative Linear function, $F(1,28) = 38.70$, $MSE = 0.58$, $p < 0.0005$, and similarly with the Positive Linear versus Quadratic comparison, $F(1,28) = 46.97$, $MSE = 0.52$, $p < 0.0005$. The Quadratic versus Negative Linear comparison was not significant, $p > 0.5$. None of the Block by function interactions were significant on these paired comparisons (although the Negative versus Positive Linear by Block comparison was significant without the Epsilon correction, $F(8,224) = 2.44$, Hunyh-Feldt Epsilon $= 0.40$, $MSE = 0.16$, $p = 0.015$, this effect disappeared when the correction was taken into account).

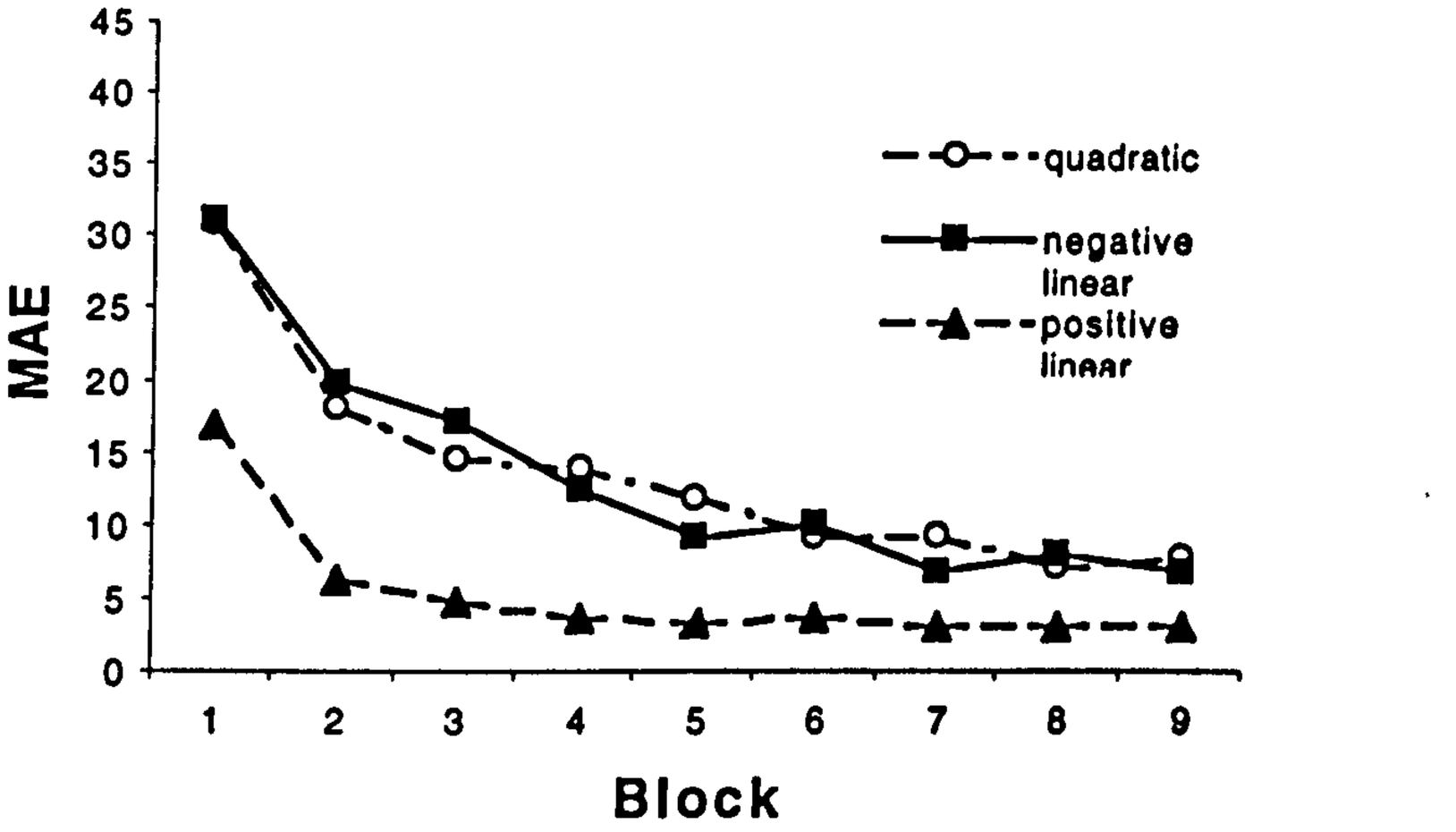


Figure 5.4 Learning curves for the functions learnt in Stage 1; either a Quadratic curve, Positive line or Negative line.

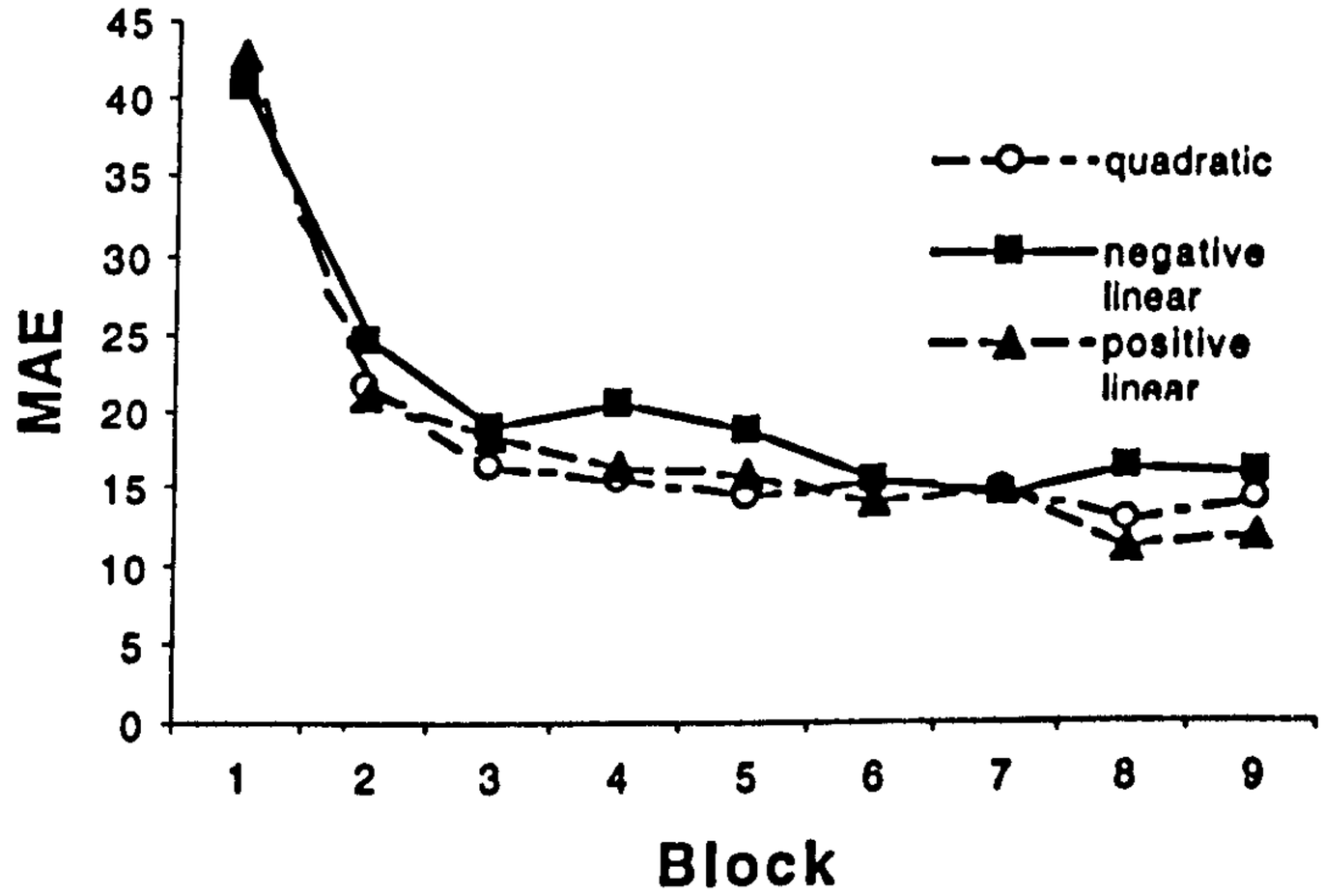


Figure 5.5 Learning curves for Stage 2. The three lines refer to the functions learnt by participants in Stage 1.

In Stage 2, there was no difference between the three function conditions, F(42,2) = 1.14, MSE = 0.55, p = 0.331, contrary to the hypotheses under consideration. There was a main effect of Block, F(8,336) = 108.53, Hunyh-Feldt Epsilon = 0.92, MSE = 0.05, p < 0.0005, but no interaction, p > 0.1.

Testing Phase

Although testing blocks were used primarily to encourage participants to abstract information rather than as opportunities to collect data, they were still analysed. In Stage 1, the main effect of Function was significant, F(2,42) = 16.23, MSE =0.13, p < 0.0005 and Block, F(1,42) = 12.82, MSE = 0.03, p < 0.001, but not the interaction, p>0.4. In Stage 2, there was also a Block effect, F(2,84) = 4.73, Hunyh-Feldt Epsilon = 1.00, p = 0.011, but no main effect of Function, F(2,42) = 1.76, MSE = 0.22, p = 0.185, or of the interaction, p>0.5. In summary then, the effects in the testing phase are identical to those found in the training phase.

Trimmed analysis

Participants may not have been able to abstract the function relating the Stage 1 input-output examples until they had learnt them to a sufficiently accurate degree. To examine this possibility, a trimmed analysis was carried out by selecting the 6 most accurate participants from each condition on the basis of their performance on the last two blocks of Stage 1. Figure 5.6 shows their MAE as a function of block and mapping. Results here were identical to that found in the full analysis. Differences arose however, when Stage 2 responses were examined. Figure 5.7 shows the Stage 2 MAE's of those 6 participants chosen

from Stage 1, for the training phase only. What looked like a very slight disadvantage for the Negative Linear group in Figure 5.5, now looks more pronounced. An ANOVA on the MAE with Block and Function as factors confirms this with a main effect of Function, $F(2,15) = 4.2$, $MSE = 0.25$, $p = 0.036$. There was no interaction between this effect and the Block effect (as above). Independent samples t-tests of the means collapsed across blocks revealed a difference between Negative Linear and Positive Linear, $t(10) = 2.41$, $p = 0.037$, but neither Negative Linear versus Quadratic, nor Positive Linear versus Quadratic proved to be reliable ($p = 0.072$ and $p = 0.87$ respectively). Performance on the testing phase revealed a main effect of Function $F(2,15) = 5.33$, $MSE = 0.07$, $p = 0.018$, but no interaction with the Block effect. Paired t-tests were again carried out, demonstrating a difference between Negative Linear and Positive Linear, $t(10) = 2.44$, $p = 0.035$, and between Negative Linear and Quadratic, $t(10) = 2.89$, $p = 0.016$.

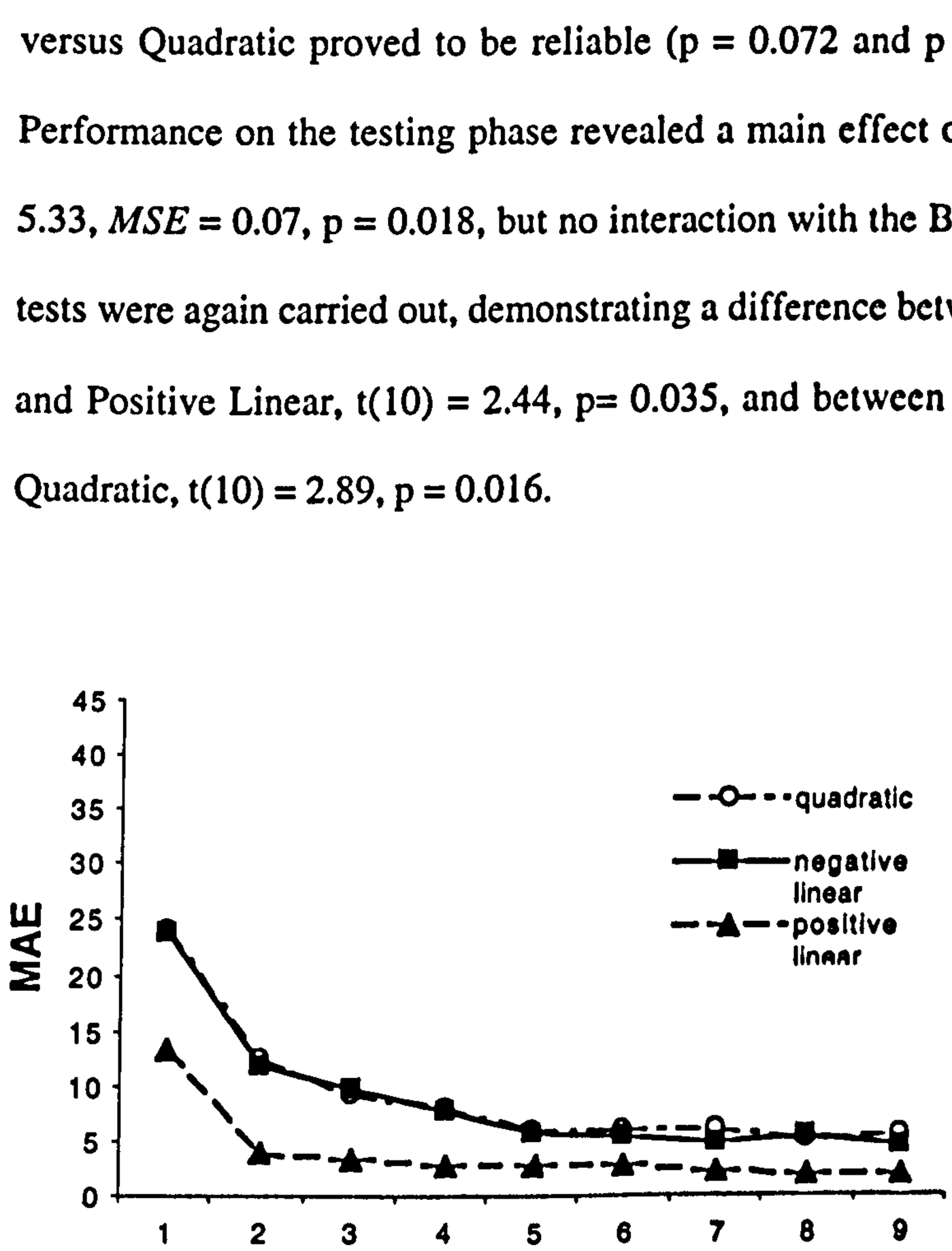


Figure 5.6 MAE as a function of Block and function type for Stage 1, trimmed participants.

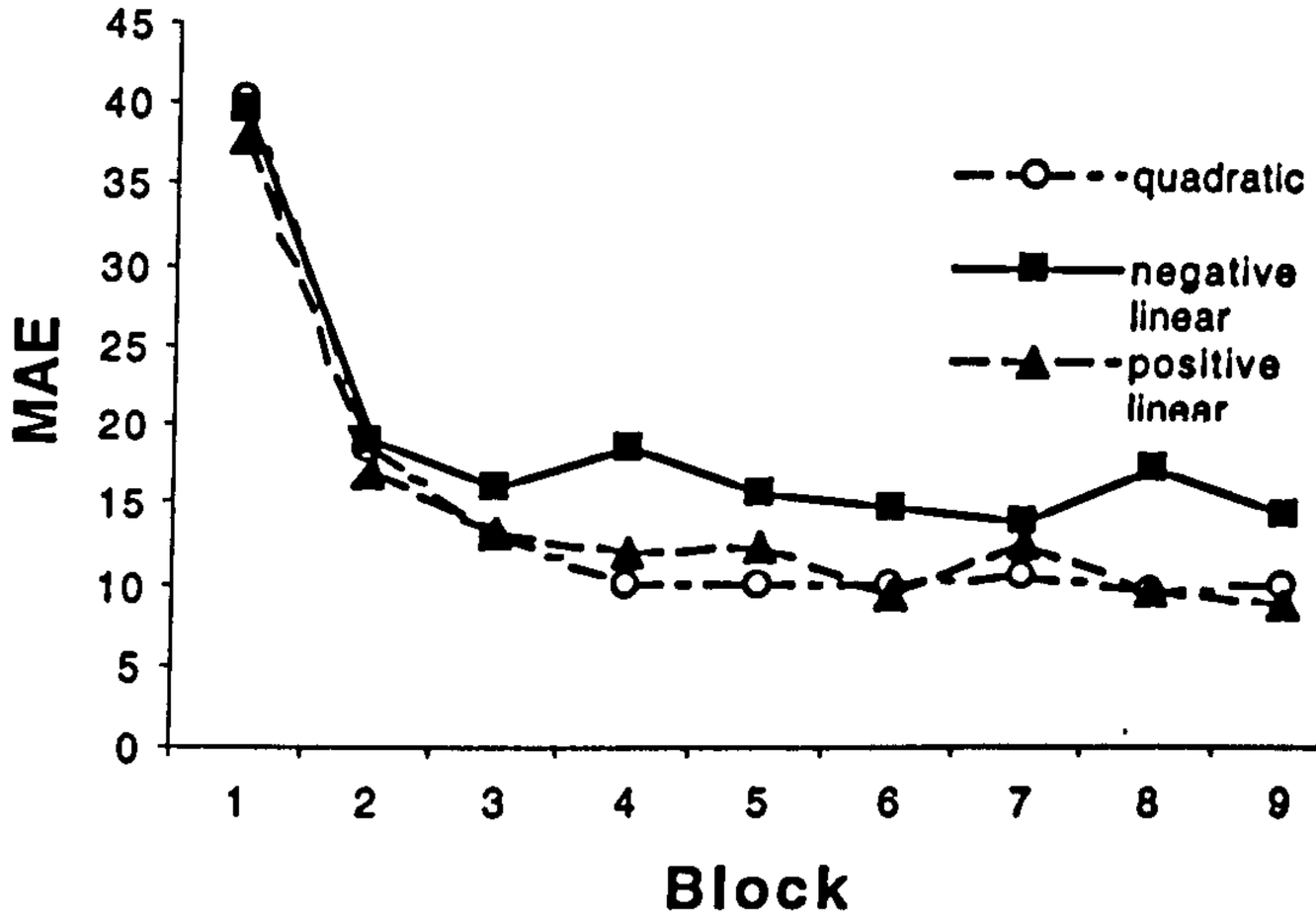


Figure 5.7 MAE as a function of Block and Stage 1 function type for Stage 2, trimmed participants.

Discussion

In the first learning task, Stage 1, it was found that participants learnt positive linear functions better than negative linear or quadratic functions, but that there was no interaction with learning block. The lack of a difference between negative linear and quadratic functions contrasts with the claim in the literature that non-monotonic functions are learnt more slowly than monotonic (in particular, Busemeyer *et al.* (1997) considers it a 'principle' of function learning, p.409). This difference is unlikely to have been a question of power because even the trimmed analysis failed to reveal any difference. Indeed the lines on Figure 5.6 are almost identical. The most likely cause are differences in methodology between this study and the previous findings. Although recent studies have used the same graphical, computer-based presentation of stimuli as was used in this

study, the test of negative linear versus quadratic functions has not been reported in the published literature. Busemeyer *et al.* (1997) cite Byun (1995) and Delosh (1995) as recent evidence, but Byun did not carry out this particular comparison and Delosh (1995) is an unpublished master's thesis. The older evidence, such as Brehmer (1974) and Sniezek and Naylor (1978) have some methodological flaws as well as paradigm differences. For instance, Sniezek and Naylor (1978) failed to randomly assign participants to different conditions (p. 371), and Brehmer's (1974) multiple comparisons are not significant when the number of comparisons are taken into account. In addition, studies from this period did not present stimuli randomly (presumably because computers were not in wide spread use) and function effects are therefore confounded with sequence effects. Further, this older work tended to present stimuli numerically rather than graphically, and use functions which were not entirely deterministic. In summary then, this result casts doubt on the generality of the claim that nonmonotonic functions are learnt more slowly than monotonic functions.

Analysis of the Stage 2 results showed no significant differences between the three function conditions when all the participants were involved, but a trimmed analysis revealed a reliable disadvantage for those who learnt a negative linear mapping in Stage 1. Disadvantages for the negative linear condition are not the results predicted from the simulations carried out using the initial weight configurations. However, they partly support a parametric account which assumes function-specific effects: those in the NL condition suffered because they were primed to expect the wrong function. It could be argued that those in the Q condition were not facilitated because they either failed to abstract the

quadratic relationship from Stage 1, or chose not to apply it in Stage 2. Further experiments can investigate whether participants are reaching the limit of their abstraction abilities with quadratic functions, or whether there is something specific about the procedures here which discourages the application of the function.

An alternative to the initial weight matrices used to generate the predictions is that participants set the weights to reflect the *proportional prior knowledge* functions suggested by Busemeyer *et al.* (1997) and described in the introduction. This would imply that participants extract the function from the first phase and apply weights which map the appropriate function between the minimum and maximum training values. From Figure 5.1, it can be seen that this would imply no difference between the PL and NL conditions because the minimum and maximum training values lie approximately on a horizontal line. As with the parametric explanation described above, the quadratic function is assumed not to have been abstracted, and thus the weight matrix not applied. This proportional weight account clearly cannot explain the findings found here. However, there is always some set of initial weights which can predict this type of order of acquisition effects. Because of this, the next experiment was designed to investigate the effects of knowledge in extrapolation, where the initial weights are far less important.

5.2 Experiment 2

Experiment 2 is again a transfer task, but the expected effects of knowledge will now be assessed by examining the generalisation patterns of participants. Specifically, if participants extrapolate in a manner which is not predicted by EXAM but is predicted by a mapping learnt in the first phase of learning, it will be argued that participants are applying a function that they acquired in the first phase.

In the Stage 1 learning task, participants are taught either a positive linear function (the Linear condition) or a function which is linear up to a point, and then flattens out (the Flat condition). Figure 5.8 indicates the stimuli presented in the experiment. Note that in Stage 1, participants in both conditions are taught the same input-output pairs until x = 70 (the vertical dashed line), but at x-values greater than 70, the target values differ for the two groups of participants. In the second stage of learning, all participants are taught on stimuli that lie on a straight line up to about half the training range of Stage 1 (shown by the line connecting empty circles in Figure 5.8) and then tested on extrapolation *x*-values.

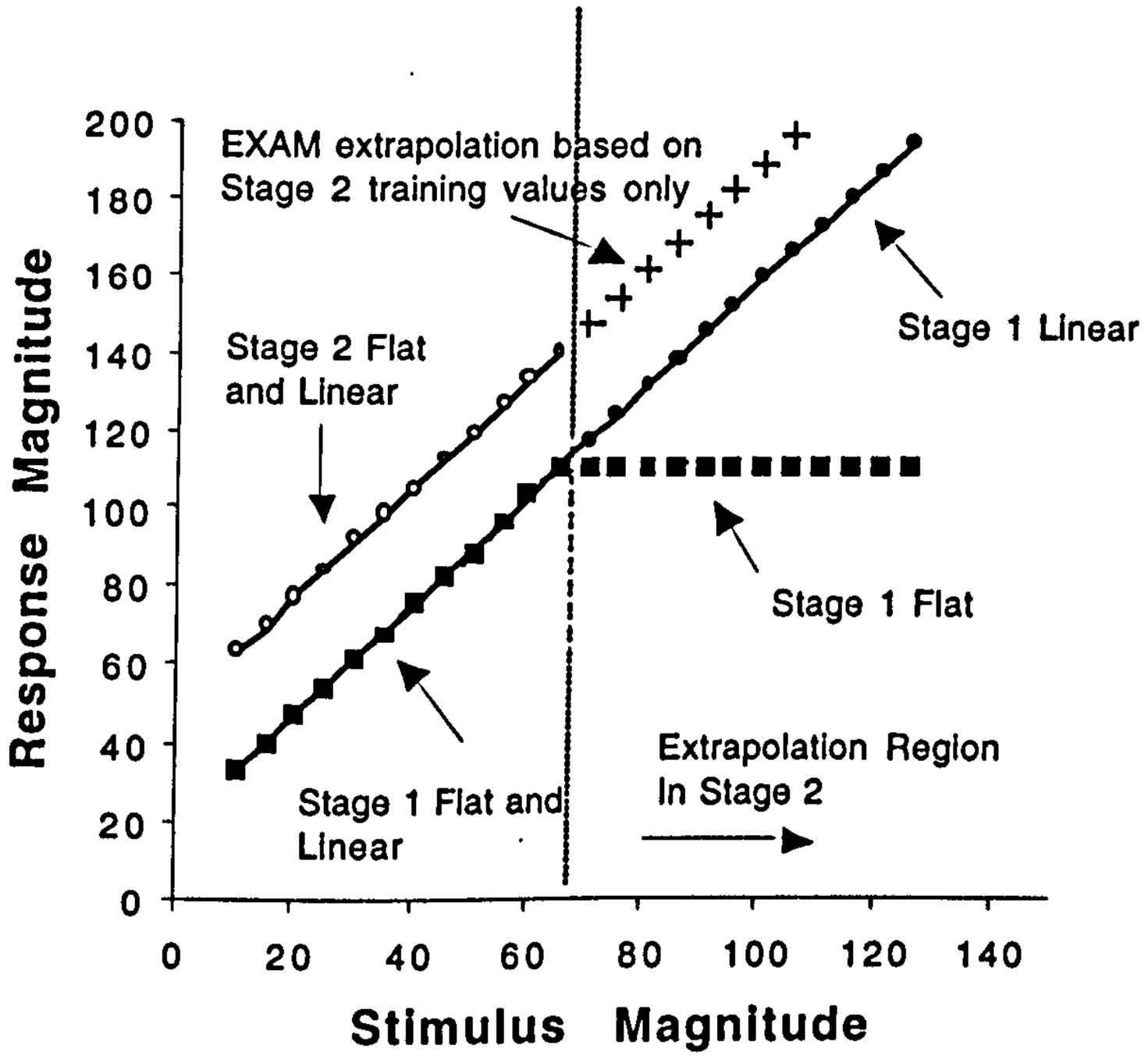


Figure 5.8 Training stimuli for Stage 1 and Stage 2. Also shown is EXAM's extrapolation pattern based on the Stage 2 training values.

The expected response pattern in this extrapolation region depends on which model participants are assumed to be using. Three possibilities are tested in this experiment. The first is a parametric model which is able to abstract a function from the Stage 1 training values and apply this function in the second stage. The second and third models are versions of EXAM adapted for a transfer task. Note that, given the Stage 2 training values, EXAM cannot restrict its range of allowable functions to include the Stage 1 extrapolation pattern – there is no λ value which predicts a constant y as x increases. This means that it has no way of *applying* the Stage 1 mapping. The three models and their predictions are described in turn.

First, consider a parametric account of the situation which predicts that participants are looking to fit an appropriate function to the Stage 2 stimuli. Because of the similarity between the Stage 1 and the Stage 2 training values, participants might be expected to abstract the function from Stage 1 and apply it in Stage 2. Applying the function should produce the same pattern of responses in Stage 2 as it did in Stage 1, that is, Stage 2 extrapolation should be approximately constant as x increases. A good computational model for their extrapolation responses would be

$$y_{S2}(x) = y_{S1}(x) + k_f \qquad (1)$$

where $y_{S2}(x)$ and $y_{S1}(x)$ are Stage 2 and Stage 1 output responses respectively, k_f is a free parameter which is constant for all x. Thus, in the Flat condition, where Stage 1 output values are constant as x increases the model predicts constant Stage 2 responses. Note that no specific parametric model of function learning is being suggested here (e.g. Brehmer, 1974; Koh & Meyer, 1991); the only requirements are that the model should be able to abstract and apply the function from Stage 1.

It could be argued that Equation 1 does not represent the application of the Stage 1 *function*, but of a *transformation of the Stage 1 output values*. The problem with answering the criticism is that the two possibilities are empirically indistinguishable within this paradigm: there is always a transformation of the old values which will generate the new responses, whether it is a simple linear one, as in this case, or a k^{th} order polynomial, which would be the case if the

extrapolation were predicted to be more complex. This will be commented on further in the Discussion section, but the important point is that a model would require some method of applying a function to perform either process: applying the appropriate transformation or the Stage 1 function. Equation 1 is meant to be agnostic on the actual algorithm involved and only includes the $y_{s,l}$ term as a means to eliminate any consistent biases participants might have in responding (which should materialise in Stage 1).

The parametric account outlined above implies that there should be differences between the extrapolation patterns of the two groups in Stage 2. Furthermore, Equation 1 should provide a better fit than other models, which are described below. Table 5.1 provides a summary of the predictions the different models, with the first row applying to the parametric account.

Data	Theory
Stage 2 Flat responses ≠ Stage 2 Linear responses Stage 2 responses best modelled with Equation 1 Stage 2 Linear gradient = Stage 1 Linear gradient	Parametric
Stage 2 Flat responses =Stage 2 Linear responses Stage 2 Linear gradient = Stage 1 Linear gradient	EXAM no-transfer
Stage 2 Flat responses ≠ Stage 2 Linear responses Stage 2 responses best modelled with Equation 2 Stage 2 Linear gradient ≠ Stage 1 Linear gradient	EXAM transfer

Table 5.1 Summary of hypotheses. All effects refer to the extrapolation region.

Two predictions could be derived for EXAM in these circumstances. To generate both of these, Stage 1 is simply a matter of optimising the model to produce the training data shown in Figure 5.8. In Stage 2, the accounts differ in terms of whether or not the input values in extrapolation region retain their mapping from the previous learning stage. One hypothesis is that participants start Stage 2 with the matrix of weights from Stage 1 and learn the new mapping in the usual way. This implies that as learning takes place, the entire set of weights will become optimised to produce the Stage 2 target values. However, because the input values in the extrapolation region are no longer receiving any kind of feedback, it becomes optimal for the model not to place any weight on those remaining basis functions and their contribution to the final solution becomes negligible. As stated in the second row of Table 5.1, this hypothesis predicts no difference between the two conditions: both extrapolation patterns are determined by EXAM's linear response rule acting on the training data of Stage 2 values (hence, this account will be referred to as EXAM's 'no-transfer' theory). Of course, there could be a large number of reasons why no transfer effects are observed, but evidence would be provided against this theory if effects did occur. Note that this account is the one closest to Busemeyer *et al.*'s (1997) explanation of prior knowledge effects in general; that is, knowledge can be incorporated by setting initial weight configurations in network.

EXAM's second account assumes that the input-output pairs from Stage 1 still exert an influence in Stage 2 extrapolation (referred to as EXAM's 'transfer' theory). This situation might arise if, for example, participants treat the two learning tasks completely separately during training and only combine them in

testing when they are less sure of the appropriate responses. Assuming some effect from the first learning stage, EXAM would predict that Stage 2 extrapolation responses will be a combination of Stage 1 output values and EXAM's linear rule (based on the Stage 2 training points). For the Flat condition, this implies that the output values will be a linearly increasing function after an initial drop (assuming constant weighting between the Stage 1 and Stage 2 mappings as x increases). This can be seen by combining the Stage 1 Flat and EXAM's Stage 2 linear extrapolation line (marked by crosses) in Figure 5.8, for x >70. Assuming there are effects from Stage 1 on the Flat condition means that there must also be effects on the mapping for the Linear condition. Thus, the linear extrapolation from Stage 2 training points should be similarly pulled down by the Stage 1 Linear mapping.

Distinguishing between a parametric account and EXAM's transfer theory is reasonably straightforward in the Linear condition: the transfer theory predicts reliable differences in the gradients of the Stage 1 and Stage 2 learning stages while the parametric account predicts no differences. In the Flat condition however, both theories predict that responses will be different to those in the Linear condition. This problem can be overcome by examining the quantitative fits of the models. Equation 1 describes the parametric model and Equation 2 represents EXAM's transfer predictions:

$$y_{S2}(x) = w \cdot y_{S2lin}(x) + (1-w) \cdot (y_{S1}(x) + k_E) \qquad (2)$$

where $y_{S2lin}(x)$ are EXAM's predictions from Stage 2 training points only, and w and k_E are free parameters. Equation 2 states that the output responses are a weighted sum of the Stage 1 values and the linear extrapolation from Stage 2 training values. Expressing the models in this way means that the parametric theory and EXAM's transfer explanation are hierarchically related; consequently, the χ^2 analysis presented in the last chapter can be used to assess whether Equation 2's extra free parameter is necessary to fit the results. The third row of Table 5.1 summarises the predictions from EXAM's transfer theory.

There were also some methodological changes between the present experiment and the last one. In addition to the instructions given in the previous experiment, participants were told in Stage 2 that it would help them to relate the new learning task back to the previous relationship. This was aimed at encouraging participants to use any functional form they had abstracted from Stage 1. A further difference between this study and Experiment 1 is that participants were trained until they reached a certain level of performance. This was to eliminate the large variability in final error scores which occurred previously.

Method

Participants

24 Warwick students were used as participants, 12 in each condition. Participants were paid £4 for the experiment, which lasted approximately 45 minutes. None had taken part in previous function learning tasks.

Design and Stimuli

In Stage 1, participants were taught to reproduce the appropriate mapping shown in Figure 5.8. The Linear mapping is given by the equation $y = 1.4x + 20$ and the flat mapping is the same up until x = 70, where y becomes constant at y = 111. For both conditions, stimulus values ranged from 10 to 125 in increments of 10. All stimuli were presented in a random order within each block. Participants continued receiving blocks of training items until they reached a criterion of a mean absolute error of 7 or less (on a scale of 0-200) on any block. After training, there were four blocks of testing where no feedback was given.

In Stage 2, all participants received training on the mapping shown by the upper solid line in Figure 5.8, given by $y = 1.4x + 50$. x varied from 10 to 70 inclusive. After participants had learnt the training data to criteria, they moved onto a test phase where they were tested on stimulus values ranging from 10 to 125 and did not receive feedback. Four blocks of testing were conducted.

As in the last experiment, stimuli and feedback were presented graphically using horizontal bars. In Stage 1, the input and output bars were red, and in Stage 2 they were yellow.

Procedure

The procedure was exactly the same as the previous experiment, with the exception that an extra line was added encouraging participants to apply any function they had acquired in the Stage 1 phase. The instructions before Stage 2 now read:

"You have now finished the first stage of the experiment. In the second half, you will learn now learn the relationship between a different drug, called Soromine, and arousal levels. Bizacol and Soromine are different drugs, but they affect the arousal level in very similar ways. It will therefore help you to try and relate the behaviour of Soromine back to the behaviour of Bizacol".

Results

11 out of 6970 Responses were removed from the analysis because their associated were 0 and reaction times less than 500 msecs.

Training data

Using the number of training blocks to reach criterion as a dependent measure, an ANOVA revealed a slight effect of Stage, $F(1,22) = 4.58$, $MSE = 16.36$, $p = 0.044$, such that Stage 2 required more blocks to learn. There were no reliable effects of the Line Slope or the interaction, p's > 0.5. Because of some extreme values in Stage 2 however, the variance in this condition was far higher. A

Wilcoxon Matched test was therefore conducted, revealing no reliable main effect of Stage. In summary, there seems to be only very small, if any, differences in the relative difficulty of the different conditions.

Testing data

Figure 5.9 shows the responses of participants in the test phase, as a function of Line Slope and Stage. The region to the right of the dotted line corresponds to the extrapolation region in Stage 2, that is, stimulus magnitudes to which participants have not received feedback. The bottom-most curve are the Stage 1 responses from participants in the Flat condition. The constant output in the extrapolation region suggests they remembered what they had been taught in the training phase. Similarly, the Stage 1 responses of those in the Linear condition appear to mirror the training values shown in Figure 5.8. Stage 2 responses are shown by the lines connecting open circles (Linear condition) and open squares (Flat condition). The extrapolation responses of participants from the Flat condition are noticeably lower than those of the Linear condition, although the responses are not as flat as those from Stage 1. This pattern suggests that there is an effect of the flat function in Stage 1, but perhaps not all participants are applying the function in Stage 2.

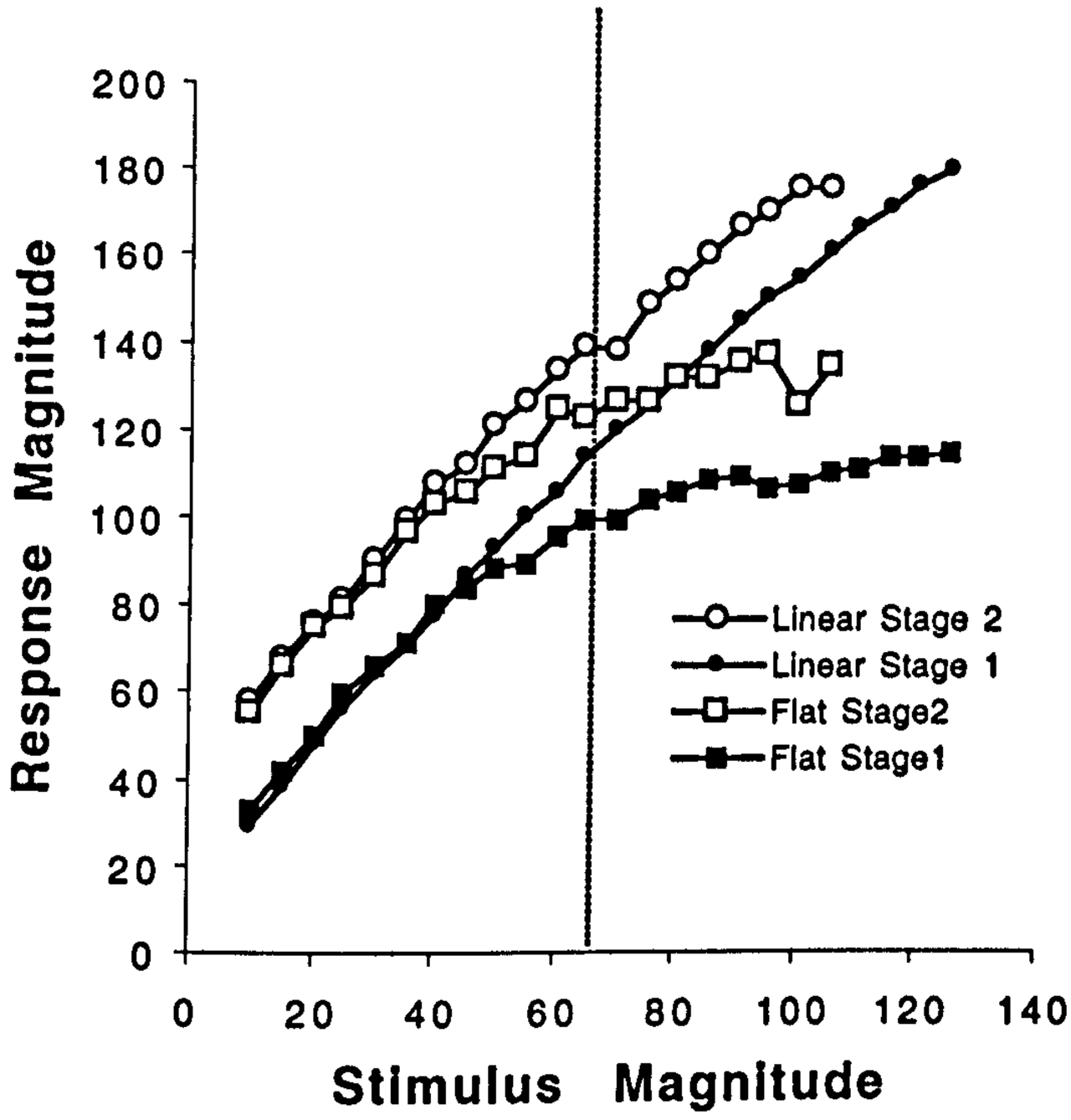


Figure 5.9 Response magnitudes as a function of Stage and type of curve learnt in Stage 1. Magnitudes to the right of the dotted line at $x = 70$ lie in the Extrapolation region.

To establish the reliability of the trends described above, an ANOVA was performed on the MAE's of participants' responses from the target values. For stimulus values participants had seen in the training phase, the target values were those presented as feedback. For both the Linear and Flat extrapolation responses, these became the *y*-values which make up the straight line continuing from the Linear Stage 2 condition, namely $y = 1.4x + 50$. As such, those participants in the Flat condition should have a higher MAE in the extrapolation region if they are attempting to fit the flat line function that they learnt in Stage 1.

The design for the analysis involved two within-subject factors with two levels: the Stage, either 1 or 2, and the Range of stimulus magnitude, either less than 70 (interpolation) or greater than 70 (extrapolation[1]); and a between subject factor, Slope, either Linear or Flat. A log transform was performed to ensure that data corresponded with the assumptions of the ANOVA. The untransformed means and standard deviations for the cells of the ANOVA are shown in Table 5.2.

There was no main effect of Slope, $F(1,22) = 0.37$, $MSE = 0.26$, $p = 0.37$, nor of the interaction between Slope and Range, $F(1,22) = 0.19$, $MSE = 0.16$, $p = 0.671$. Stage 2 had reliably higher MAE's overall than Stage 1, $F(1,22) = 39.08$, $MSE = 0.19$, $p < 0.0005$ and there was an interaction between Slope and Stage, $F(1,22) = 5.46$, $MSE = 0.19$, $p = 0.029$, such that in Stage 2, those in the Flat condition had a higher MAE than those in the Sloped condition, as predicted by the hypothesis. Not surprisingly, there was also a main effect of Range, $F(1,22) = 22.77$, $MSE = 0.16$, $p < 0.0005$, an interaction of Range and Stage, $F(1,22) = 21.56$, $MSE = 0.05$, $p < 0.0005$, and an interaction of Range by Stage by Slope, $F(1,22) = 25.56$, $MSE = 0.05$, $p < 0.0005$. The effects of Range indicate that the differences of interest all took place in the extrapolation region.

	stage 1		stage 2	
	interpolation	extrapolation	interpolation	extrapolation
Linear	6.63	10.86	9.52	15.16
	(5.34)	(1.75)	(8.01)	(2.56)
Flat	7.64	6.93	10.74	28.80
	(1.22)	(1.50)	(17.52)	(3.24)

Table 5.2 Means of the untransformed MAE of participants' responses from the target value. Standard deviations in parentheses.

[1] In Stage 1, there is no extrapolation range (participants are only tested on magnitudes on which they have had feedback). This term is used for its relevance to Stage 2 and to complete the design.

Analysis of gradients

The first hypothesis to be considered is whether participants from the Stage 2 Linear group had significantly different patterns of responses from those in the Stage 1 Linear group. This is a test of EXAM's transfer theory, as indicated by the third row of Table 5.1. The analysis was carried out by determining the gradients of the linear regression lines for Stage 1 and 2 Linear responses. These were found to be 1.08 and 1.03 with standard errors 0.068 and 0.075 respectively. These were based on 382 individual responses (12 participants with 32 data points each) for the Stage 1 learning task and 380 for the Stage 2 task (11 participants with 32 data points each and 1 participant with 30 data points, the other two having been removed as outliers). A 2-tailed t-test revealed no significant differences between the two gradients, $t(768) = 0.49$, $p > 0.5$, which provides evidence against EXAM's transfer theory. An individual participant analysis was also conducted on the gradients of each participant in the Stage 2 Linear condition, versus the pooled gradient from those in the Stage 1 Linear condition. This also revealed no significant differences between the two conditions.

Although the ANOVA revealed that there was a significant difference between those in the Stage 2 Linear group and those in Stage 2 Flat group, high variance in the Flat condition cell suggests large individual differences. This possibility was assessed by comparing the gradients of those in the Stage 2 Flat condition with those in the Stage 2 Linear condition, for the extrapolation region. Table 5.3 shows the gradients for the individual participants in the Flat condition based

on 32 data points. The t-value is the test statistic describing the difference between the gradient for that particular participant and the gradient for the Stage 2 Linear condition, distributed on $N_1 + N_2 - 4 = 406$ degrees of freedom. When adjusted for the number of comparisons being made, these must produce a p-value of less than 0.005 to be significant at the $\alpha = 0.05$ level. As can be seen from the p-value column, 5 of the participants do not differ significantly and 2 have a reliably higher gradient than in the Linear condition. The fact that the 5 remaining participants extrapolate with lower gradients than the Stage 2 Linear condition suggests support for either the parametric theory, or EXAM's transfer theory (as stated in the first and third rows of Table 5.1).

Participant	b	standard error	t-value	p-value
1	-0.200	0.260	-4.581	0.000
2	1.500	0.099	3.909	0.000
3	1.190	0.165	0.892	0.373
4	0.358	0.266	-2.452	0.015
5	0.399	0.113	-4.795	0.000
6	1.600	0.157	3.327	0.001
7	1.010	0.262	-0.077	0.938
8	0.498	0.158	-3.100	0.002
9	0.896	0.107	-1.065	0.287
10	0.926	0.177	-0.554	0.580
11	0.257	0.075	-7.655	0.000
12	-0.236	0.146	-7.870	0.000

Table 5.3 Gradients of the regression line through the responses in the extrapolation region of Stage 2. The 12 participants are from the FL slope condition. The t-scores and associated p-values indicate whether each individual participants's gradient differs to that of the pooled responses from the PL condition.

Model fitting

To determine which of these theories provides the better account of the data, Equations 1 and 2 were fitted to the average responses for the individual participants in the Stage 2 Flat condition. These equations are repeated below for convenience:

$$y_{S2}(x) = y_{S1}(x) + k_f \tag{1}$$

$$y_{S2}(x) = w \cdot y_{S2lin}(x) + (1-w) \cdot (y_{S1}(x) + k_E) \tag{2}$$

$y_{S1Flat}(x)$ and $y_{S2Lin}(x)$ were calculated by using the linear regression of the appropriate scores. These were found to be $y_{S1Flat} = 0.216x + 88.47$ and $y_{S2Lin} = 1.08x + 67.21$. Optimisation of the free parameters k_f, w, and k_E was carried out by minimising the summed squared error (SSE) between the models' predictions and participants' responses. Table 5.4 displays the results.

Participant	k_f	w	k_E	χ^2	p-value
1	19.71	0.00	19.71	0.00	1.0000
2	67.74	1.00	-	14.88	0.0001
3	73.65	1.00	-	18.43	0.0000
4	30.24	0.16	25.49	0.23	0.6309
5	19.27	0.21	9.84	3.49	0.0617
6	61.40	1.00	-	7.82	0.0052
7	35.65	0.66	0.00	9.91	0.0016
8	50.08	0.32	47.94	5.64	0.0176
9	34.33	0.63	0.00	11.92	0.0006
10	54.77	0.82	55.85	11.36	0.0008
11	9.24	0.05	7.01	1.05	0.3047
12	17.55	0.00	17.55	0.00	1.0000

Table 5.4 Results of fitting Equations 1 and 2 to the responses of those in the Flat condtion, Stage 2 extrapolation region. Columns 2, 3, and 4 refer to the best-fitting parameter values (dashes indicate that the parameter is irrelvant to determining final fit). The fourth column is the χ^2 value obtained from Equation 20, Chapter 4, with N =8. The fifth column is the p-value associated with the model comparisions.

Columns 2, 3 and 4 reveal the best-fitting parameter values for the parametric theory (Equation 1) and EXAM's transfer theory (Equation 2) respectively. High values of the w parameter indicate that the responses from these participants are best determined from the Stage 2 Linear group's regression line; low values of w mean that responses are best predicted from the Stage 1 Flat responses. The p-values in column 6 indicate to whether or not there is a significant advantage for the model with the extra free parameter (EXAM's transfer theory). As before, these values must be below 0.005 to be significant at

the 0.05 level. There are several important conclusions to be drawn from the model comparisons. First, of the 5 participants who were identified as having significantly different gradients from the Stage 2 Linear condition, none of them require any weight on the Stage 2 Linear component in Equation 2. This is demonstrated by the non-significant χ^2 value for Participants 1, 5, 8, 11, and 12. Participants 4 is also best modeled with Equation 1, which ties in with his low p-value in the previous analysis. These results confirm that at least half the participants are best modeled by assuming that they abstracted a function from Stage 1, and applied it in Stage 2.

Of the six remaining participants, three had a w parameter value of 1.00. This means that there was no effect of the Stage 1 training values at all. These provide evidence against Exam's transfer theory, but support the no-transfer version. Finally, three participants had significant p-values with some effect of both Stage 1 training values and Stage 2 linear extrapolation, supporting Exam's transfer theory.

Discussion

This experiment addressed the question of how transfer effects in extrapolation could be accounted for. One hypothesis put forward was that there would no transfer effects - participants would eliminate any initial weight configurations they had left over form Stage 1, and simply learn the Stage 2 data, as predicted by one implementation of EXAM. This experiment has provided conclusive

evidence against this: at least half the participants in the Flat condition showed reliably different patterns of responding to those in the Linear condition.

Another hypothesis was that participants retain the associations they had learnt in Stage 1 and produce responses based partly on the Stage 1 values, and partly on the extrapolation predicted by EXAM from the Stage 2 training values. Evidence was provided against this by the finding that, in the Stage 2 Linear condition, no participants showed any effects of the Stage 1 target values (gradients of responses between the two learning stages were not reliably different). Furthermore, in the Flat condition, only 3 out of the 12 participants required both components to predict the data. This finding indicates that although some participants may use this strategy, the majority are performing the task in a different way.

Finally, it was argued that participants might abstract a function from Stage 1, and apply this function in Stage 2. Applying the function should produce the same pattern of responses in Stage 2 as it did in Stage 1, that is, Stage 2 extrapolation should be entirely predictable from a parameterised function of Stage 1 responses. All participants from the Linear condition and 6 from the Flat condition are consistent with this view. In summary, people used a variety of strategies when performing the extrapolation in the second stage. However, it is clear that a model must have some method with which to apply a function to the learning situation if it is to account for all of the behaviour demonstrated in this experiment.

In the introduction it was suggested that an alternative to applying a function was to perform a transformation on the Stage 1 training values. For example, to generate the Stage 2 response to $x = 100$, a Flat condition participant might retrieve from memory the appropriate Stage 1 output of $y_{s1} = 110$, and then add the appropriate transformation value of 30 to produce $y_{s2} = 140$. There are several points to be made about this explanation. First, applying a transformation like $y_{s2}(x) = y_{s1}(x) + 30$ to all the input values (including the extrapolation range) requires abstracting this transformation function from the Stage 1 data and applying it to the Stage 2 task – a procedure that requires the same mechanisms as the original parametric account. Thus, a model which doesn't have the capability of representing functions, like EXAM, would fail to reproduce the behaviour whichever theory was found to be responsible. Secondly, basing responses on the Stage 1 output values requires keeping the original mapping in memory while learning the new one. Although this is not completely implausible, it is a far less efficient process than directly abstracting the function. Furthermore, it becomes increasingly less efficient as more and more example mappings of the same function are observed.

Designing an experiment to separate the two would have to involve moving outside the current methodology. As the introduction mentioned, any behaviour in the extrapolation region of Stage 2 could be explained by some transformation of the Stage 1 output values. One possibility might be to have the Stage 2 learning task in a completely different domain, for example using tones of different durations. However, participants could map the rectangle space onto the tone space, transfer training exemplars, and perform the extrapolation on the

basis of these. Of course, the 'work' for the system is in performing the transformation appropriately, as it is in this experiment, but it still might be possible to argue that the function from Stage 1 has not been extracted.

Two conclusions should be drawn on the issue of transformation versus function abstraction. First, EXAM has no mechanism for performing a transformation or the application of a function from Stage 1. Secondly, because the transformation account is a less efficient approach to performing the Stage 2 extrapolation, the onus should be placed on finding empirical data demonstrating that participants use the transformation procedure, rather than the other way around.

5.3 General Discussion

The two experiments presented in this chapter have examined the predictions of Delosh *et al.*'s (1997) models concerning transfer effects. The EXAM and the ALM assume that all effects of prior knowledge are incorporated into the initial weight matrix of the models, which encourages certain weight solutions to be found over others. Experiment 1 investigated whether participants would show an advantage on a second learning task if they were taught a similar function in a first task. In Stage 1 of this experiment, it was found that participants showed indistinguishable performance when learning a negative linear or a quadratic function, but that they were better on a positive linear function. Given the previous work has suggested that monotonic functions are learnt more slowly than non-monotonic functions, this result is a useful contribution to the area. In Stage 2, the results demonstrated a reliable disadvantage for those who saw a negative linear function in the first stage of learning. This was interpreted as support for a parametric account of function learning. It was assumed that those in the NL condition suffered because they were predicting the wrong type of function to be learnt.

Experiment 2 examined the effect prior knowledge has on generalisation patterns. Some participants were able to abstract a function in the Stage 1 learning task, and apply that function to their Stage 2 extrapolation responses. Two versions of how EXAM might perform in a transfer task were discussed, but neither was able to reproduce the patterns of responses better than a parametric generalisation strategy.

Taken together, these experiments imply that models of function learning must have the following attributes: (1) the representational capacity of the model must be such that it can restrict its range of allowable solutions to psychologically appropriate forms (2) the learning algorithm must show a deficit if it assumes the wrong functional form; (3) the model must be able to accurately learn previously unencountered patterns of training values; and (4) it must have some mechanism with which to transform these patterns into an abstract form. Experiment 1 provided evidence for (1) and (2) by showing that that participants in the NL condition observe that negative linear pattern in the Stage 1 learning phase, apply it to the Stage 2 task and perform worse on the task as a result. Experiment 2 required that they learn the training stimuli in Stage 1, abstract this function, and apply it in Stage 2. This provides evidence for (3) and (4).

None of the models in the literature have all of these capabilities. First, EXAM has been shown to be lacking (1), (2) and (4). Secondly, the parametric models of Brehmer (1974) and Koh and Meyer (1991), discussed in the previous chapter, have problems accounting for our ability to learn patterns of responses which do not conform to simple polynomial function in x – if a high order polynomial is fitted to the training data (so that they can reproduce the data points accurately), the extrapolation patterns of the model do not conform to those of the participants (see Delosh *et al*, 1997). This means they fail to account for (3) and consequently (4). Brehmer specified that participants search through a hierarchy of polynomials, which implies that initially searching for the wrong model would slow learning and predicts (2). Koh and Meyer suggest participants expand their

representation as learning continues (through a regularisation term) which again implies learning deficits if the wrong starting representation is assumed. The model that was developed in the previous chapter, RERM, fails to specify a learning algorithm to test (2) and has no mechanism for moving from exemplars to rules. Further work can focus on how to augment the current models to satisfy these criteria.

5.4 Conclusions

This chapter has demonstrated the importance of prior knowledge used for its information value. To *apply* a function is to *restrict* the range of possible solutions the algorithm is capable of fitting. The most successful function learning model to date, the EXAM (Delosh *et al.*, 1997), has been shown to be incapable of restricting its functions appropriately.

Chapter 6

The aim of this thesis was to examine the effects of prior knowledge on categorisation and function learning abilities. This chapter draws together the findings presented here and discusses the possibilities for future work within the area.

Summary and discussion of results

Chapter 2 discussed the need to incorporate knowledge into models of learning and how current psychological models might be augmented to do so. The chapter starts off by describing Geman, Bienenstock and Doursat's (1992) distinction between *bias* and *variance*. It was shown that when estimating a function from a set of examples, there are two factors which might lead the generalisation error to be high. First, it might be case that the estimated function varies substantially with the individual data sets. This was referred to as the variance component. Secondly, the estimated function might be far away from the true function, averaged across all the different data sets - the bias. It was argued that eliminating both bias and variance was impossible for models which are highly empirically driven, such as Nosofsky's (1986) GCM; there simply aren't enough exemplars of the concept in the environment. The solution to this is to incorporate appropriate prior knowledge into these models. Knowledge about the generating function means that incorrect solutions can be eliminated,

thus reducing variance. Bias is not increased, however, because only the incorrect solutions have been removed.

The concept of knowledge as being a reduction in the range of allowable solutions for an algorithm is a very important one. If it is accepted that learning systems must restrict their possible functions in some way, it can be seen that the organism cannot just focus on how to associate attributes of the environment, but it must in some way determine what *not* to learn. In other words, prior knowledge plays as vital a role in determining the behaviour of the system as data driven learning does.

Using prior knowledge to alleviate the bias / variance dilemma involves using knowledge for its information value. Chapter 2 also demonstrated that learning can benefit from knowledge which reduces the complexity of the task (Abu-Mostafa, 1995). Complexity knowledge allows the organism to reach the solution to a problem more efficiently. For example, knowing that the optimum level for a weight is around the middle of its range means that fewer iterations are needed to arrive at a solution, although generalisation performance is not necessarily improved.

The distinction between information and complexity was used to breakdown the notion of prior knowledge into well defined sub-components. Some psychological and statistical findings can now be seen as being concerned with information, while others are better thought of as referring to the complexity of the problem. Furthermore, the gap between psychology and statistics was

bridged by suggesting how statistical approaches might be used to model prior knowledge findings.

The third chapter presented modelling and experimental work based on several experiments carried out by Heit and Bott (2000). Heit and Bott examined how the appropriate prior knowledge was selected and applied to the learning of new concepts. We demonstrated that participants map their known concepts onto new categories, thereby facilitating them on both presented and unpresented features which conform to the structure of the known concept.

The results from Heit and Bott (2000) were simulated using a modular network with different modules corresponding to different types of prior knowledge. This approach was based on the committees of networks suggested by Jacobs, Jordan, Nowlan and Hinton (1991), as discussed in Chapter 2. When the experiments of Heit and Bott were simulated, the network gradually learnt to apply a known category distinction to the new learning task. The reason the model learnt to apply this particular category distinction is that this one proved useful - other category distinctions were not beneficial to the network and were therefore not allocated weight. Once learning was completed, it was able to classify unpresented instances by using its prior category knowledge combined with the newly learnt mapping.

As well as modelling Heit and Bott (2000), extra simulations were carried out to generate predictions from the model. The most important of these was the demonstration that blocking effects arise from using useful knowledge in a

learning task. For example, if the network was provided with hints concerning the mapping between the known concept and the classes of the new one, there was an improvement on those features which were aligned with the prior category, but a depreciation in performance for those which were not. However, an experiment conducted to test this prediction failed to find the effects. Although more experiments are required before it can be concluded that blocking doesn't occur in this experimental situation, the lack of an effect serves to highlight the fact that people do not simply decide what to learn on the basis of which module reduces training error the most. The danger of overfitting the data was discussed extensively in Chapter 2 - these findings demonstrate that people use other methods as well as the training error to resolve the problem of how much to pay attention to the data.

The simulations made several important contributions to prior knowledge research. First, they demonstrated that there are advantages in modelling the effects of knowledge - original and testable predictions can be generated from such attempts. Secondly, they illustrated that modelling knowledge effects requires more than simply configuring the initial weights of a purely empirical network. The modular approach demonstrated here was shown to capture the pattern of responses in the Heit and Bott (2000) experiments and to be a statistically sensible approach to incorporating knowledge. As such, this method looks a promising start for modelling these kinds of effects.

Finally, these simulations allowed links to be drawn between the modular network used here and similar networks used in the categorisation literature by

Erickson and Kruschke (1998). Erickson and Kruschke modelled the interaction between a rule-based (uni-dimensional) classification system, and an exemplar-based one using the mixture of experts architecture (Jacobs *et al.*, 1991). The use of the same kind of modelling system implies that there may also be similarities in the underlying problem structure between the two domains. In this case, both situations involve a highly flexible classification system (exemplar and empirically driven modules) combined with a more rigid system (rules and prior knowledge modules) and the psychological problem is to determine how these components interact.

Chapters 4 and 5 continued this line of thought by treating the application of known continuous functions as the application of prior knowledge, compared with more flexible, non-parametric regression algorithms in function learning tasks. The suggestion put forward by Delosh, Busemeyer and McDaniel (1997) was that a non-parametric representation combined with a linear extrapolation rule (the EXAM model) was sufficient to explain the way people generalise when learning continuous input-output mappings. These two chapters demonstrated that this was not the case - knowledge plays a far bigger role in determining extrapolation behaviour.

Chapter 4 provided the first example of non-monotonic extrapolation in function learning. This was achieved by training participants on a cyclic curve and then examining their extrapolation patterns. It was found that participants continue to the nonmonotonic mapping in a cyclic manner, regardless of whether they received instructions consistent with a cyclic relationship or a more neutral cover

story. A data-driven model like EXAM cannot explain this result - a model with some way of applying a cyclic function is required. Another important finding was that participants were shown to move from a linear extrapolation at the start of learning, towards a cyclic response pattern as learning progressed. In other words, there was an increasing effect of knowledge in a similar way to the participants in the Heit and Bott (2000) experiments and the Baywatch model.

The chapter also presented and fitted a model which combined a rule-component and an exemplar component (known as RERM), in a similar manner to Erickson and Kruschke's (1998) model described above. The model highlighted an interesting question for function learning and prior knowledge in general. Fitting RERM demonstrated the difficulty in modelling the progression from the flexible, exemplar module, towards the more rigid cyclic rule module, as participants did. Erickson and Kruschke's model solved the problem by allocating attention to the component which predicted the data best. They then demonstrated that as learning progressed, the model was able to discover which parts of the input space were best controlled by each module in terms of the resulting prediction error. However, this approach was shown not to be possible for RERM because the exemplar system has the capability to reduce training error more than the cyclic module, regardless of the area of the input space (this is another way of saying that the exemplar system is more flexible). Several possible solutions to this were discussed in the chapter, but it was concluded that participants must have some concept of the problems of over-fitting a data set to avoid exclusively using the exemplar component.

The fifth chapter examined in more detail what it means to learn a function parametrically, and whether EXAM has the capability to represent functions in the same way that participants do. To apply a particular function means to restrict the range of allowable solutions to those belonging to that class. Although EXAM and many other non-parametric models have the capability to restrict their range of solutions - through the smoothing parameter - they are not able to restrict them in a psychologically plausible way.

This was demonstrated by two transfer tasks. In Experiment 1, participants were taught training values consistent with either a positive linear function, a quadratic function, or a negative linear function in the first stage of learning. In the second stage of learning, all participants were taught a quadratic function. Those who learnt the negative linear function first were reliably worse in the second stage of learning. This was interpreted as evidence that they were initially restricting themselves to the wrong type of function - something they could only do if the algorithm they were using was capable of restricting its allowable solutions to a negative linear function. The second experiment involved training participants on a particular pattern of responses in the first phase, and testing whether they could apply that function with different parameter values in the second stage. At least half the participants exhibited behaviour consistent with the idea that they had abstracted and applied the new function.

The experiments presented in Chapter 5 confirmed that models of function learning must have the capability to restrict their solutions, but also that they can acquire new functions in some way. None of the models published so far have

proposed mechanisms by which people might be able to store appropriate patterns and then apply this prior knowledge at a later date.

Future directions

When discussing how prior knowledge might be deconstructed in Chapter 2, it was argued that the effects of prior knowledge should be classed as either providing extra *information* to the organism, or reducing the *complexity* of the task. However, modelling the effects of prior knowledge in psychology has focused to a far greater extent on the complexity issues, for example by incorporating initial weight configurations (Busemeyer, Byun, Delosh, & McDaniel, 1997). Chapters 2 and 5 suggested that, although complexity issues are important in some situations, there are several reasons why more of an emphasis should be placed on the role of information. These can be summarised by reiterating the fact that questions concerning information can be pitched at a far more general level than those about complexity – information knowledge should benefit any algorithm, whereas complexity knowledge has to be tuned to suit each algorithm independently. This implies that psychologists concentrating on complexity issues risk more error because of large amounts of variability in the different algorithms participants use within and between experiments.

So, how should psychologists proceed to investigate the information effects of prior knowledge? The answer to this lies in investigating the role of the smoothing parameter in models of categorisation and regression. As evidence for this claim, note that the smoothing parameter has played a vital role in the

discussion of prior knowledge throughout this thesis: the bias / variance distinction in Chapter 2; combining predictions from multiple hypotheses in Chapter 3; the progression from an exemplar module to rule module in Chapter 4; and in Chapter 5's discussion of the need to restrict models in psychologically plausible ways. Given that there have been very few studies investigating how people set the level of smoothing, this is a clear direction for future research.

References

Abu-Mostafa, Y. S. (1993). Hints and the VC dimension. *Neural Computation, 5*, 278-288.

Abu-Mostafa, Y. S. (1995). Hints. *Neural Computation, 7*, 639-671.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*, 716-723.

Allen, S. W., & Brooks, L. R. (1991). Specializing the operation of an explicit rule. *Journal of Experimental Psychology: General, 120*, 3-19.

Anderson, J. R., & Finchman, J. M. (1996). Categorization and sensitivity to correlation. *Journal of Experimental Psychology: Learning, Memory and Cognition, 22*, 259-277.

Ashby, F. G., & Alfonso-Reese, L. A. (1995). Categorization as probability density estimation. *Journal of Mathematical Psychology, 39*, 216-233.

Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U., & Waldron, E. M. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review, 105*, 442-481.

Ashby, F. G., & Gott, R. E. (1988). Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology-Learning Memory and Cognition, 14*, 33-53.

Ashby, F. G., & Lee, W. W. (1991). Predicting similarity and categorization from identification. *Journal of Experimental Psychology-General, 120*, 150-172.

Ashby, F. G., & Maddox, W. T. (1993). Relations between prototype, exemplar, and decision bound models of categorization. *Journal of Mathematical Psychology, 37*, 372-400.

Ashby, F. G., & Townsend, J. T. (1986). Varieties of perceptual independence. *Psychological Review, 93*, 154-179.

Barnard, E., & Casasent, D. (1991). Invariance and neural nets. *IEEE Transactions on Neural Networks, 3*, 232-240.

Barsalou, L. W., & Huttenlocher, J. (1998). Basing categorization on individuals and events. *Cognitive Psychology, 36*, 203-272.

Bedford, F. L. (1989). Constraints on learning new mappings between perceptual dimensions. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 232-248.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour.* New Jersey: Princeton University Press.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

Borowiak, D. (1989). *Model discrimination for nonlinear regression models.* New York: Marcel Dekker.

Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organization Behaviour and Human Performance, 11*, 1-27.

Brehmer, B., Kuylenstierna, J., & Liljergen, J.-E. (1974). Effects of function form and cue validity on the subjects' hypotheses in probabilistic inference tasks. *Organizational behaviour and human performance, 11*, 338-354.

Brown, G. D. A., Hulme, C., Hyland, P. D., & Mitchell, I. J. (1994). Cell suicide in the developing nervous system: a functional neural network model. *Cognitive Brain Research*, 1994.

Busemeyer, J., McDaniel, M. A., & Byun, E. (1997). The abstraction of intervening concepts from experience with multiple input-multiple output causal environments. *Cognitive Psychology, 32*, 1-48.

Busemeyer, J. R., Byun, E., Delosh, E., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks. In K. Lamberts & D. Shanks (Eds.), *Knowledge, Concepts and Categories* . Hove, East Sussex: Psychology Press.

Byun, E. (1995). *Interaction between type of non-linear relationship on function learning.* , Purdue Univeristy.

Chater, N. & Brown, G.D. A. (1999). Scale-invariance as a unifying psychological principle. *Cognition 69(3),* 17-24.

Caroll, J. D. (1963). *Functional learning: the learning of continuous functional mappings relating stimulus and response continua* (RB-63-26). Princeton, New Jersey: Educational Testing Service.

Choi, S., McDaniel, M. A., & Busemeyer, J. R. (1993). Incorporating prior biases in network models of conceptual rule learning. *Memory & Cognition, 21*, 413-423.

Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General, 120*, 235-253.

Delosh, E. L. (1999). Recognition of exceptions and rule-consistent items in the function learning domain. *Proceedings of the Twenty First Annual Conference of the Cognitive Science Society*, Vancouver.

DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology-Learning Memory and Cognition, 23*, 968-986.

Dienes, Z., Altman, G., & Gao, S.-J. (in press). Mapping across domains without feedback. *Cognitive Science*.

Duda, R. A., & Hart, E. H. (1973). *Pattern Classification and Scene Analysis*. New York; Chichester: Wiley-Interscience.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition, 48*, 71-99.

Ennis, D. M. (1988). Confusable and discriminable stimuli: Comment on Nosofsky (1986) and Shepard (1986). *Journal of Experimental Psychology: General, 117*, 408-411.

Erickson, M. A., & Kruschke, J. K. (1998). Rules and exemplars in category learning. *Journal of Experimental Psychology: General, 127*, 107-140.

Estes, W. K. (1984). Global and local control of choice behavior by cyclically varying outcome probabilities. *Journal of Experimental Psychology: Learning, Memory and Cognition, 10*, 258-270.

Estes, W. K. (1994). *Classification and cognition*. New York: Oxford University Press.

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing* (Vol. 2, pp. 524-532). San Mateo, CA: Morgan Kaufman.

Flannagan, M. J., L. S. Fried, & Holyoak, K. J. (1986). Distributional expectations and the induction of category structure. *Journal of Experimental Psychology: Learning, Memory and Cognition, 12*, 241-256.

Frasconi, P., Gori, M., & Soda, G. (1995). Recurrent neural networks and prior knowledge for sequence processing: a constrained nondeterministic approach. *Knowledge-Based-Systems, 8*, 313-328.

Fukushima, K. (1988). Neocognitron: a hierarchical neural network capable of visual pattern recognition. *Neural Networks, 1*, 119-130.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation, 4*, 1-58.

Giles, C. L., & Omlin, C. W. (1993). Extraction, insertion and refinement of symbolic rules in dynamically driven recurrent neural networks. *Connection Science, 5*, 307-337.

Gluck, M. A., & Bower, G. H. (1988). From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General, 117*, 227-247.

Goldstone, R. L. (1994). Influences of categorization on perceptual discrimination. *Journal of Experimental Psychology: General, 123*, 178-200.

Goldstone, R. L., Steyvers, M., Spencer-Smith, J., & Kersten, A. (1999). Interactions between perceptual and conceptual learning. In E. Diettrich & A. Markman (Eds.), *Cognitive Dynamics: Conceptual Change in Humans and Machines* . Cambridge, MA: MIT Press.

Hahn, U., & Chater, N. (1997). Concepts and similarity. In K. Lamberts & D. Shanks (Eds.), *Knowledge and concept learning* (pp. 43-92). London: Psychology Press.

Hahn, U., & Chater, N. (1998). Similarity and rules: distinct? exhaustive? empirically distinguishable? *Cognition, 65*, 197-230.

Hampton, J. (1997). Conceptual combination. In K. Lamberts & D. Shanks (Eds.), *Knowledge, concepts, and categories* (pp. 133-160). London: Psychology Press.

Hanson, S. J., & Pratt, L. Y. (1989). Comparing biases for minimal network construction with back-propagation. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing* (Vol. 1, pp. 177-185). San Mateo, CA: Morgan Kaufman.

Haussler, D. (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence, 36*, 177-221.

Hayes, B. K., & Taplin, J. E. (1992). Developmental changes in categorization processes: Knowledge and similarity-based models of categorization. *Journal of Experimental Child Psychology, 54*, 188-212.

Heit, E. (1994). Models of the effects of prior knowledge on category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20*, 1264-1282.

Heit, E. (1995). Belief revision in models of category learning. *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, Pittsburgh.

Heit, E. (1997). Knowledge and concept learning. In K. Lamberts & D. Shanks (Eds.), *Knowledge, concepts, and categories* (pp. 7-41). London: Psychology Press.

Heit, E. (1998). Influences of prior knowledge on selective weighting of category members. *Journal of Experimental Psychology: Learning, Memory and Cognition, 24*, 712-731.

Heit, E., & Bott, L. A. (2000). Knowledge selection in category learning. *The Psychology of Learning and Motivation, 39*, 163-199.

Homa, D. (1984). On the nature of categories. *Psychology of Learning and Motivation-Advances in Research and Theory, 18*, 49-94.

Jacobs, R. A. (1995). Methods for combining experts probability assessments. *Neural Computation, 7*, 867-888.

Jacobs, R. A. (1997). Nature, nurture, and the development of functional specializations: A computational approach. *Psychonomic Bulletin & Review, 4*, 299-309.

Jacobs, R. A., Jordan, M. I., & Barto, A. G. (1991). Task decomposition through competition in a modular connectionist architecture. The what and where vision tasks. *Cognitive Science, 16*, 219-250.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation, 3*, 79-87.

Kaplan, A. S., & Murphy, G. L. (2000). Category learning with minimal prior knowledge. *Journal of Experimental Psychology: Learning Memory and Cognition, 26*, 829-846.

Koh, K., & Meyer, D. E. (1991). Function learning: induction of continuous stimulus-response relations. *Journal of Experimental Psychology: Learning, Memory & Cognition, 17*, 811-836.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22-44.

Lamberts, K. (1994). Flexible tuning of similarity in exemplar-based categorization. *Journal of Experimental Psychology-Learning Memory and Cognition, 20*, 1003-1021.

Lamberts, K. (1995). Categorization under time pressure. *Journal of Experimental Psychology-General, 124*, 161-180.

Lamberts, K. (1997). Process models of categorization. In K. Lamberts & D. Shanks (Eds.), *Knowledge, Concepts and Categories* (pp. 371-403). Hove: Psychology Press.

Lamberts, K., & Shapiro, L. (in press). Exemplar models and category-specific deficits. In E. M. E. Forde & E. W. Humphreys (Eds.), *Category-specificity in brain and mind* : Psychology Press.

Land, E. H. (1964). The retinex. *American Scientist, 52*, 247-264.

Marechsal, D., & Shultz, T. R. (1996). Generative connectionist networks and constructivist cognitive development. *Cognitive development, 11*, 571-603.

Mareschal, D., & Shultz, T. R. (1993). A connectionist model of the development of seriation. *Proceedings of the Fifteenth Annual*

Conference of the Cognitive Science Society. Hillsdale, NJ: Erlbaum.

Markman, E. M. (1990). Constraints children place on word meanings. *Cognitive Science, 14*, 57-77.

McCloskey, M., & Cohen, N. J. (1986). Catastrophic interference and connectionist networks: The sequential learning problem. In G. Bower (Ed.), *The psychology of learnining and motivation* (Vol. 24, pp. 109-165). San Diego, CA: Academic Press.

McKinley, S. C., & Nosofsky, R. M. (1995). Investigations of exemplar and decision bound models in large, ill-defined category structures. *Journal of Experimental Psychology-Human Perception and Performance, 21*, 128-148.

Medin, D. L., & Ortony, A. (1989). Psychological essentialism. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 179-195). Cambridge: Cambridge University Press.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review, 85*, 207-238.

Minsky, M., & Papert, S. (1969). *Perceptrons*. Cambridge, MA.: MIT Press.

Moody, J. E., & Darken, C. J. (1989). Fast learning in networks of locally tuned processing units. *Neural Computation, 1*, 281-294.

Mozer, M. C., & Smolensky, P. (1989). Using relevance to reduce network size automatically. *Connection Science, 1*, 3-16.

Murphy, G. L., & Allopenna, P. D. (1994). The locus of knowledge effects in concept learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20*, 904-919.

Murphy, G. L., & Kaplan, A. S. (2000). Feature distribution and background knowledge in category learning. *Quarterly Journal of Experimental Psychology, 53A*, 962-982.

Murphy, G. L., & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review, 92*, 289-316.

Murphy, G. L., & Wisniewski, E. J. (1989). Feature correlations in conceptual representations. In G. Tiberghien (Ed.), *Advances in Cognitive Science* (Vol. 2, pp. 23-45). Chichester: Ellis Horwood.

Naylor, J. C., & Clark, R. D. (1968). Intuitive inference strategies in interval learning tasks as a function of magnitude and sign. *Organizational Behaviour and Human Performance, 3*, 378-399.

Naylor, J. C., & Domine, R. K. (1981). Inference based on uncertain data: Some experiments on the role of slope magnitude, instructions, and stimulus distribution shape on the learning of contingency relationships. *Organizational Behaviour and Human Performance, 27*, 1-31.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General, 115*, 39-57.

Nosofsky, R. M. (1988a). Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 700-708.

Nosofsky, R. M. (1988b). Similarity, frequency, and category representations. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*, 54-65.

Nosofosky, R. M. (1988c). On exemplar-based representations: Reply to Ennis

(1988). *Journal of Experimental Psychology: General, 117*, 412-414.

Nosofsky, R. M. (1990). Relations between exemplar-similarity and likelihood models of classification. *Journal of Mathematical Psychology, 34*, 393-418.

Nosofsky, R. M., Kruschke, J. K., & McKinley, S. C. (1992). Combining exemplar-based category representations and connectionist learning rules. *Journal of Experimental Psychology-Learning Memory and Cognition, 18*, 211-233.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review, 101*, 53-79.

Nosofsky, R. M., & Zaki, S. R. (1998). *Dissociations between categorization and recognition in amnesics and normals: An exemplar-based interpretation* (Research Report 213): Cognitive Science Program, Indiana University.

Omlin, C. W., & Giles, C. L. (1996). Rule revision with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering, 8*, 1992.

Pazzani, M. J. (1991). Influence of prior knowledge on concept acquisition: Experimental and computational results. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 17*, 416-432.

Pearce, J. M. (1987). A model for stimulus generalization in Pavlovian conditioning. *Psychological Review, 94*, 61-73.

Perrone, M. P. (1994). General averaging results for convex optimisation. In M. C. Mozer et al. (Eds.), *Proceedings of the 1993 Connectionist Models Summer School*. Hillsdale, NJ: Lawrence Erlbaum.

Piaget. (1965). *The child's concept of number.* New York: Norton Library.

Pinker, S., & Prince, A. (1988). On language and connectionism - analysis of a parallel distributed-processing model of language-acquisition. *Cognition, 28*, 73-193.

Poggio, T., Torre, V., & Koch, C. (1985). Computational vision and regularisation theory. *Nature, 317*, 314-319.

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology, 77*, 353-363.

Posner, M. I., & Keele, S. W. (1970). Retention of abstract ideas. *Journal of Experimental Psychology, 83*, 304-308.

Prechelt, L. (1997). Investigation of the CasCor Family of Learning Algorithms. *Neural Networks, 10*, 885-896.

Quartz, S. R., & Sejnowski, T. J. (1997). The neural basis of cognitive development: A constructionist manifesto. *Brain and Behavioural Sciences, 20.*

Rehder, B. (1999). A causal-model theory of categorization. *Proceedings of the Twenty First Annual Conference of the Cognitive Science Society,* Vancouver, Canada, August 19-21, 1999.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical Conditioning II: Current Research and Theory* . New York: Appleton-Century-Crofts.

Ripley, B. D. (1996). *Pattern recognition and neural networks.* Cambridge: Cambridge University Press.

Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology, 7*, 573-605.

Rosseel, Y. (1998). *Categorization as probability density estimation: statistical and computational models of categorization and category learning.* , University of Gent, Gent.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propogation. In D. E. Rummelhart, J. L. McClelland, & t. P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1: Foundationa, pp. 318-362). Cambridge, MA: MIT Press.

Salatas, H., & Bourne, L. E. (1974). Learning conceptual rules: III. Processes contributing to rule difficulty. *Memory & Cognition, 2*, 549-553.

Sawyer, J. E. (1991). Effects of risk and uncertainty on judgements of contingency relations and behavioral resource allocation decisions. *Orgnaizational Behavior and Human Performance, 49*, 124-150.

Schmidhuber, J. (1997). Discovering neural nets with low kolmogorov complexity. *Neural Networks, 10*, 857-873.

Shultz, T. R., Buckingham, D., & Oshima-Takane, Y. (1994). A connectionist model of the learning of personal pronouns in English. In S. J. Hanson, T. Petsche, M. Kearns, & R. L. Rivest (Eds.), *Computational learning theory and natural learning systems* (Vol. 2). Cambridge, MA: MIT Press.

Shultz, T. R., & Schmidt, W. C. (1991). A cascade-correlation model of balance scale phenomena. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Schyns, P. G., Goldstone, R. L., & Thibaut, J. P. (1998). The development of features in object concepts. *Behavioral and Brain Sciences, 21*, 1-35.

Schyns, P. G., & Rodet, L. (1997). Categorization creates functional features. *Journal of Experimental Psychology-Learning Memory and Cognition, 23*, 681-696.

Shanks, D. R. (1991). Categorization by a connectionist network. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 17*, 433-443.

Shepard. (1988). Time and distance in generalization and discrimination: Reply to Ennis (1988). *Journal of Experimental Psychology: General, 117*, 415-416.

Shepard, R. N. (1989). Internal representation of internal regularities: A challenge for connectionism. In L. Nadel, L. A. Cooper, P. Culicover, & R. M. Harnish (Eds.), *Neural Connections, Mental Computation* (pp. 104-134). Cambridge: MA, London: England: Bradford/MIT Press.

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorisation of classifications. *Psychological Monographs, 75*, Whole No. 517.

Sniezek, J. A., & Naylor, J. C. (1978). Cue measuement scale and functional hypothesis testing in cue probability learning. *Organizational Behaviour and Human Decision Processes, 22*, 366-374.

Suddarth, S., & Holden, A. (1991). Symbolic neural systems and the use of hints for developing complex systems. *International Journal of Machine Studies, 35*, 291.

Towell, G. G., Shavlik, J. W., & Noordewier, M. O. (1990). Refinement of aproximate domain theories by knowledge-based neural networks. *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston. MA. USA.

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*, 327-352.

Vandierendonck, A. (1995). A parallel rule activation and rule synthesis model for generalization in category learning. *Psychonomic Bulletin & Review, 2*, 442-459.

Vandierendonck, A., & Rosseel, Y. (2000). Interaction of knowledge-driven and data-driven processing in category learning. *European Journal of Cognitive Psychology, 12*, 37-63.

Vapnik, V., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications, 16*, 264-280.

Waganaar, W. A., & Sagaria, S. D. (1975). Misperception of exponential growth. *Perception and Psychophysics, 18*, 416-422.

Waldmann, M. R., & Holyoak, K. J. (1992). Predictive and diagnostic learning within causal models: Asymmetries in cue competition. *Journal of Experimental Psychology: General, 121*, 222-236.

Waldmann, M. R., Holyoak, K. J., & Fratianne, A. (1995). Causal models and the acquisition of category structure. *Journal of Experimental Psychology: General, 124*, 181-206.

Ward, T. B. (1993). Processing biases, knowledge, and context in category formation. In G. V. Nakamura, R. Taraban, & D. L. Medin (Eds.), *Categorization by Humans and Machines* (pp. 257-282). San Diego: Academic Press.

Wattenmaker, W. D., Dewey, G. I., Murphy, T. D., & Medin, D. L. (1986). Linear separability and concept learning: Context, relational properties, and concept naturalness. *Cognitive Psychology, 18*, 158-194.

Westerman. (2000). Modelling cognitive development with constructivist neural networks. *Proceedings of the Sixth Neural Computation and Psychology Workshop*, Liege, Belgium,

Wisniewski, E. J. (1995). Prior knowledge and functionally relevant features in concept learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*, 449-468.

Wisniewski, E. J., & Medin, D. L. (1991). Harpoons and long sticks: The interaction of theory and similarity in rule induction. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning* . San Mateo, CA: Morgan Kaufman.

Wisniewski, E. J., & Medin, D. L. (1994). On the interaction of theory and data in concept learning. *Cognitive Science, 18*, 221-282.